

การทำนายกำลังผลิตของโซลาร์เซลล์โดยใช้โมเดลประมาณค่าและตัวกรองคาลมาน



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Photovoltaic Power Generation Forecasting by using Estimator Model and Kalman
Filter



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2019

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การทำนายกำลังผลิตของโซลาร์เซลล์โดยใช้โมเดลประมาณค่าและตัวกรองคาลมาน
โดย	นายพีรพล จิรนนทเจริญ
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร.วาทีต เบญจพลกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วิทยากร อัครวริเศษ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.วาทีต เบญจพลกุล)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุรัชย์ ชัยทัศน์)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.ภูมิพัฒน์ แสงอุดมเลิศ)

พีรพล จิรนนท์เจริญ : การทำนายกำลังผลิตของโซลาร์เซลล์โดยใช้โมเดลประมาณค่าและตัวกรองคาลมาน. (Photovoltaic Power Generation Forecasting by using Estimator Model and Kalman Filter) อ.ที่ปรึกษาหลัก : ศ. ดร.วาทิต เบญจพลกุล

วิทยานิพนธ์ฉบับนี้เสนอแนวทางในการทำนายกำลังผลิตของโซลาร์เซลล์โดยใช้แบบจำลอง Auto Regressive Integrated Moving Average (ARIMA) และ Kalman filter algorithm แนวทางในการทำนายนี้ทำนายกำลังผลิตแบบเวลาจริงของสถานี Photovoltaic (PV) ใดๆที่ต้องการทราบโดยในวิทยานิพนธ์ฉบับนี้จะทำนายกำลังผลิตทุกๆ 5 นาที จุดมุ่งหมายของการใช้ Kalman filter algorithm เพื่อติดตามกำลังผลิตแบบเวลาจริงของสถานีที่ต้องการทราบค่ากำลังผลิตในกรณีที่มีข้อมูลกำลังผลิตขาดหายไปในช่วงเวลา Kalman filter มีข้อดีที่ไม่ต้องใช้ข้อมูลในอดีตจำนวนมากในการทำนายและติดตามข้อมูลที่ต้องการ แต่เนื่องจาก Kalman filter algorithm ต้องการข้อมูลการวัดค่าแบบเวลาจริงเพื่อปรับแก้ในสมการดังนั้นเราจึงได้เสนอแบบจำลองการคำนวณค่าประมาณกำลังผลิต Estimator model เพื่อคำนวณค่ากำลังผลิตแบบเวลาจริงเพื่อนำไปใช้ใน Kalman filter algorithm ซึ่ง Estimator model จะคำนวณค่าประมาณกำลังผลิตแบบเวลาจริงของสถานีที่เราต้องการทำนายจากข้อมูลค่ากำลังผลิตแบบเวลาจริงของสถานีข้างเคียง จากนั้นจึงส่งผลลัพธ์นั้นไปใช้ในการทำนายกำลังผลิตจาก Kalman filter algorithm พบว่าผลลัพธ์ที่ได้มีความใกล้เคียงกับค่าจริงโดยสามารถดูได้จากค่าดัชนีชี้วัดความแม่นยำ นอกจากนี้เรายังเปรียบเทียบผลลัพธ์ของการทำนายกับ Persistence Model และ Artificial Neural Network ซึ่งผลลัพธ์ปรากฏว่า ARIMA-Kalman ให้ผลลัพธ์ที่ดีกว่า Persistence Model และมีความแม่นยำใกล้เคียงกับ Artificial Neural Network ที่ใช้กันมากขึ้นเรื่อยๆในปัจจุบัน แต่ ARIMA-Kalman มีข้อดีตรงที่ไม่ต้องใช้ข้อมูลกำลังผลิตของสถานีที่ต้องการทำนายกำลังผลิตเป็นจำนวนมากในการทำนาย

สาขาวิชา วิศวกรรมไฟฟ้า

ลายมือชื่อนิสิต

ปีการศึกษา 2562

ลายมือชื่อ อ.ที่ปรึกษาหลัก

6070263921 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: Real time forecast, Auto Regressive Integrated Moving Average,
Kalman filter algorithm, Photovoltaic forecast

Peeraphon Jiranantacharoen : Photovoltaic Power Generation Forecasting
by using Estimator Model and Kalman Filter. Advisor: Prof. Watit
Benjapolakul, D.Eng.

This thesis presents an approach to forecast power generation of PV by using the Auto-Regressive Integrated Moving Average model (ARIMA) and Kalman filter algorithm. This approach forecasts power generation of the interested PV every 5 minute interval. The purpose of Kalman filter algorithm is to track the real-time power generation of the interested PV, especially when we cannot receive real-time power generation data from the interested PV. The advantage of Kalman Filter algorithm is the fact that it can perform, even if we do not have many historical data. However, Kalman filter algorithm requires real-time measurement data to adjust the forecast value, therefore, we propose the Estimator model to estimate real-time power generation of the interested PV by using information of real-time power generation from neighboring PV. Then, we send the result to Kalman filter algorithm and forecast power generation. The forecasting result shows that it is close to the real data value and we calculate model accuracy to support the forecasting result. Furthermore, we compare the result with Persistence Model and Artificial Neural Network. The comparison result shows that ARIMA-Kalman is better than Persistence Model. The accuracy is nearly the same as that of Artificial Neural Network, but it can forecast without the interested PV's historical data.

Field of Study: Electrical Engineering

Student's Signature

Academic Year: 2019

Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ ศาสตราจารย์ ดร. วาทิต เบญจพลกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำแนะนำ และให้การสนับสนุนการวิจัยเป็นอย่างดีตลอดมา ทำให้นิสิตมีความรู้ ความเข้าใจทฤษฎีและปฏิบัติมากยิ่งขึ้น ผู้วิจัยจึงขอขอบพระคุณไว้ ณ ที่นี้

ขอขอบคุณผู้ช่วยศาสตราจารย์ ดร.วิทยากร อัครวิเศษ ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.สุรัชย์ ชัยทัศนีย์ และ รองศาสตราจารย์ ดร.ภูมิพัฒน์ แสงอุดมเลิศ กรรมการสอบวิทยานิพนธ์ ที่ได้สละเวลาตรวจสอบ ให้คำแนะนำ และตอบคำถามทุกคำถาม เพื่อให้ผู้วิจัยเรียนรู้ถึงรายละเอียดของวิทยานิพนธ์ทุกประเด็น อีกทั้งยังทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น และขอขอบคุณอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ อบรม สั่งสอน จนทำให้ข้าพเจ้ามีความรู้ ความสามารถในการทำงาน และดำรงชีวิตในสังคมได้อย่างมีความสุข

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยที่ให้ ความรู้และประสบการณ์ดี ๆ ทั้งด้านวิชาการ ด้านสังคมและอื่นๆแก่ข้าพเจ้า

ขอขอบคุณพระคุณ บิดา-มารดา และครอบครัวเป็นอย่างสูงสำหรับความช่วยเหลือในทุกๆ ด้าน และคอยให้กำลังใจ

ขอขอบคุณ Dr. Ngoc Thien Le รุ่นพี่ร่วมอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่คอยชี้แนะและให้ คำปรึกษาในความรู้และเนื้อหาที่เกี่ยวข้องกับวิทยานิพนธ์มาโดยตลอด รวมทั้ง ให้คำปรึกษาเมื่อเกิดปัญหาใน การทำงาน จนกระทั่งสามารถทำวิทยานิพนธ์สำเร็จได้ด้วยดี

ขอขอบคุณพี่ๆ เพื่อนๆที่ ภาควิชาวิศวกรรมไฟฟ้าที่คอยให้กำลังใจ คำปรึกษา และความมี น้ำใจช่วยเหลือซึ่งกันและกัน จนทำให้ผู้วิจัยสามารถทำวิทยานิพนธ์ได้สำเร็จสมบูรณ์

พีรพล จิรนนทเจริญ

สารบัญ

	หน้า
.....	ค
บทคัดย่อภาษาไทย.....	ค
.....	ง
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ญ
สารบัญรูปภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของวิทยานิพนธ์.....	1
1.2 งานวิจัยที่เกี่ยวข้อง.....	2
1.3 วัตถุประสงค์ของวิทยานิพนธ์.....	6
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	6
1.5 เนื้อหาของวิทยานิพนธ์.....	7
บทที่ 2 แนวเหตุผลและทฤษฎีที่เกี่ยวข้อง.....	8
2.1 Time series Analysis.....	8
2.1.1 Introduction.....	8
2.1.2 Autocorrelation Function.....	8
2.1.3 Linear Stationary Model.....	10
2.1.4 Linear Non-Stationary Model.....	11
2.1.5 Box & Jenkins Methodology.....	14

2.1.6 Model Identification & Model Estimation.....	15
2.1.7 Model Diagnostic Checking	15
2.2 Kalman Filter.....	17
2.2.1 State Forecast	17
2.2.2 Filtered States	18
2.3 Artificial Neural Network (ANN).....	21
2.4 การทดสอบค่าความแม่นยำของผลลัพธ์จากแบบจำลอง	29
2.4.1 Root Mean-Square Error (RMSE).....	29
2.4.2 Normalized Root Mean-Square Error (nRMSE)	29
2.4.3 Mean Absolute Percentage Error (MAPE).....	30
2.4.4 Mean Square Error (MSE).....	30
2.4.5 Skill Score.....	30
บทที่ 3 แนวทางที่เสนอ	33
3.1 ภาพรวมขั้นตอนการทำงาน	33
3.2 การเก็บข้อมูล	35
3.3 การออกแบบ Estimator Model	39
3.4 การใช้ ARIMA Model เพื่อสร้าง State Space Model.....	42
3.5 การทำนายโดยใช้ Kalman Filter.....	47
บทที่ 4 การทดสอบและผลลัพธ์	49
4.1 การทดสอบผลลัพธ์ของการเลือกใช้ชุดข้อมูลที่ต่างกันในการ fit ARIMA Model.....	49
4.1.1 กรณีใช้ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศแจ่มใส	51
4.1.2 กรณีใช้ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศมีเมฆมาก	52
4.1.3 กรณีใช้ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศฝนตก	53
4.1.4 กรณีที่ใช้วันทั้ง 3 แบบข้างต้นเพื่อทำนายกำลังผลิตเป็นเวลา 1 เดือน	54

4.1.5	สรุปผลการทดสอบ	55
4.2	Case study 1: 3 Solar PV System	56
4.2.1	Data Inspection	56
4.2.2	Calculating Power Generation by using Estimator Model	57
4.2.3	Created ARIMA Model	58
4.2.4	Forecast without Estimator Model (Observation data is available).....	61
4.2.5	Forecast with Estimator Model (Observation data is unavailable)	62
4.2.6	Forecasting by Artificial Neural Network.....	62
4.2.7	Forecasting Result	65
4.2.8	สรุปผลการทดสอบ	65
4.3	Case study 2: 5 Solar PV Systems.....	67
4.3.1	Data Inspection	67
4.3.2	Calculating Power Generation by using Estimator Model	68
4.3.3	Created ARIMA Model	70
4.3.4	Forecast without Estimator Model (Observation data is available).....	71
4.3.5	Forecast with Estimator Model (Observation data is unavailable)	72
4.3.6	Forecasting by Artificial Neural Network.....	72
4.3.7	Forecasting Results	75
4.3.8	สรุปผลการทดสอบ	75
4.4	การทดสอบผลของระยะทางต่อความแม่นยำของ Estimator Model	77
4.4.1	ระยะไม่เกิน 10 กิโลเมตร	77
4.4.2	ระยะระหว่าง 10 กิโลเมตรถึง 20 กิโลเมตร	77
4.4.3	ระยะทางตั้งแต่ 20 กิโลเมตรขึ้นไป	80
4.4.4	สรุปผลการทดสอบ	84

บทที่ 5	สรุปผลการทดสอบ	85
5.1	บทสรุป	85
5.2	ข้อเสนอแนะ	87
บรรณานุกรม	88
ประวัติผู้เขียน	90



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญตาราง

	หน้า
ตารางที่ 2-1 ความสัมพันธ์ของค่า p , q กับกราฟ ACF, PACF	15
ตารางที่ 2-2 อัลกอริทึมที่ใช้ในการ train แบบจำลอง	24
ตารางที่ 4-1 ตารางชี้วัดความแม่นยำในกรณีใช้ข้อมูลวันที่อากาศแจ่มใสในการสร้างแบบจำลองเพื่อทำนายกำลังผลิตในวันที่ 15/3/2018.....	51
ตารางที่ 4-2 ตารางชี้วัดความแม่นยำในกรณีใช้ข้อมูลวันที่อากาศมีเมฆมาก	52
ตารางที่ 4-3 ตารางชี้วัดความแม่นยำในกรณีใช้ข้อมูลวันที่มีลักษณะสภาพอากาศฝนตก ในการสร้างแบบจำลองเพื่อทำนายกำลังผลิตในวันที่ 15/3/2018.....	53
ตารางที่ 4-4 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิต ของสถานี Apy ในวันที่ 15/3/2018.....	65
ตารางที่ 4-5 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิต ของ PV2 ในวันที่ 29/10/2019	75
ตารางที่ 4-6 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิต ของ PV2 ในวันที่ 29/10/2019 ในระยะทางระหว่าง 10-20 กิโลเมตร.....	80
ตารางที่ 4-7 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิตของ PV2 ในวันที่ 29/10/2019 ในระยะทางมากกว่า 20 กิโลเมตร.....	82

สารบัญรูปภาพ

	หน้า
รูปที่ 2-1 Flowchart กระบวนการของ Box & Jenkins	14
รูปที่ 2-2 Flow chart แสดงการทำงานของ Kalman filter.....	20
รูปที่ 2-3 โครงสร้างทั่วไปของ Artificial Neural Network.....	21
รูปที่ 2-4 ตัวอย่าง block diagram ของ open loop (รูปบน) และ closed loop (รูปล่าง).....	22
รูปที่ 2-5 การแบ่งข้อมูลที่ป้อนเข้าไปเพื่อ train แบบจำลอง	23
รูปที่ 2-6 ขั้นตอนระบุจำนวน neuron และ delays ของระบบ	24
รูปที่ 2-8 อินเทอร์เฟซสรุปภาพรวมของระบบหลังการ train	26
รูปที่ 2-7 อินเทอร์เฟซเลือก training algorithm และการ train แบบจำลอง.....	26
รูปที่ 2-9 ตัวอย่างของ performance plot.....	27
รูปที่ 2-10 ตัวอย่าง regression plot ของ Validation set.....	28
รูปที่ 2-11 อินเทอร์เฟซการสร้าง script.....	28
รูปที่ 3-1 ภาพรวมการทำงานของระบบ	34
รูปที่ 3-2 homepage ของ www.pvoutput.org	35
รูปที่ 3-3 แถบเครื่องมือของเว็บไซต์.....	35
รูปที่ 3-4 หน้าต่างเลือกประเทศที่ต้องการเพื่อดูข้อมูลของสถานีโซลาร์เซลล์	36
รูปที่ 3-5 หน้าต่างรายชื่อสถานีโซลาร์เซลล์ที่มีข้อมูลในประเทศที่เลือก	36
รูปที่ 3-6 ตารางแสดงรายชื่อและระยะทางของสถานีรอบข้างกับสถานีที่เลือก	37
รูปที่ 3-7 ตารางแสดงข้อมูลกำลังผลิตและพลังงานที่โซลาร์เซลล์ผลิตได้ของสถานีที่เลือกที่เวลาต่างๆ ใน 1 วันโดยข้อมูลจะอัปเดตทุก 5 นาที	38
รูปที่ 3-8 กราฟเปรียบเทียบค่าประมาณที่ได้จากโมเดลประมาณค่าและค่าที่ได้จากการวัดจริงของ สถานีทั้ง 3 สถานี.....	41

รูปที่ 3-9 รูปชุดข้อมูลตัวอย่างของกำลังผลิตทั้งหมด 30 วันของสถานี Apy ในเดือน มีนาคม พ.ศ. 2561.....	43
รูปที่ 3-10 รูปชุดข้อมูลตัวอย่างหลังการทำ differencing 1 ครั้ง.....	44
รูปที่ 3-11 รูปแสดงการกราฟ ACF ของชุดข้อมูลตัวอย่าง.....	44
รูปที่ 3-12 รูปแสดงการกราฟ PACF ของชุดข้อมูลตัวอย่าง	45
รูปที่ 3-13 รูปแสดงรายละเอียดค่าสัมประสิทธิ์และค่าพารามิเตอร์หลังจากที่สร้างแบบจำลอง ARIMA(0,1,1).....	46
รูปที่ 4-1 แผนที่แสดงที่ตั้งของ solar rooftop ทั้ง 3 แห่ง	49
รูปที่ 4-2 ตัวอย่างรูปกำลังผลิตในวันที่ท้องฟ้าแจ่มใส	51
รูปที่ 4-3 รูปกำลังผลิตขณะสภาพอากาศมีเมฆมาก	52
รูปที่ 4-4 รูปกำลังผลิตขณะสภาพอากาศมีฝนตก	53
รูปที่ 4-5 ผลลัพธ์ของค่า nRMSE เมื่อใช้วันที่มีสภาพอากาศฝนตก, แจ่มใส และ มีเมฆมากทำนาย กำลังผลิตของวันที่มีสภาพอากาศฝนตก.....	54
รูปที่ 4-6 ผลลัพธ์ของค่า nRMSE เมื่อใช้วันที่มีสภาพอากาศฝนตก, แจ่มใส และ มีเมฆมากทำนาย กำลังผลิตของวันที่มีสภาพอากาศมีเมฆมาก	55
รูปที่ 4-7 กำลังผลิตของสถานี Apy ในวันที่ 12/3/2018.....	56
รูปที่ 4-8 รูปภาพเปรียบเทียบค่าประมาณที่ได้จากสมการของ Estimator model กับค่าจริง.....	57
รูปที่ 4-9 ค่าเฉลี่ยของ correlation ในเวลา 1 เดือน	57
รูปที่ 4-10 ACF plot และ PACF plot ของกำลังผลิตสถานี Apy	58
รูปที่ 4-11 กำลังผลิตหลังจาก difference 1 ครั้งของ Apy ในวันที่ 12/3/2018	58
รูปที่ 4-12 ACF plot และ PACF plot ของกำลังผลิตสถานี Apy หลังจาก difference 1 ครั้ง.....	58
รูปที่ 4-13 Residual plot ของ ARIMA(4,0,0).....	59
รูปที่ 4-14 Residual plot ของ ARIMA(0,1,4).....	60
รูปที่ 4-15 parameter ของ ARIMA(4,0,0)	60

รูปที่ 4-16 paramater ของ ARIMA(0,1,4)	60
รูปที่ 4-17 Residual Histogram และ ACF plot ของ ARIMA(4,0,0)	61
รูปที่ 4-18 Residual Histogram และ ACF plot ของ ARIMA(0,1,4)	61
รูปที่ 4-19 กำลังผลิตของสถานี Apy ในวันที่ 15/3/2018 จากการทำนายที่เป็นผลลัพธ์จาก Kalman filter โดยใช้ข้อมูลกำลังผลิตแบบเวลาจริง	61
รูปที่ 4-20 กำลังผลิตของสถานี Apy ในวันที่ 15/3/2018 จากการทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator model คำนวณค่าประมาณข้อมูลกำลังผลิตแบบเวลาจริง	62
รูปที่ 4-21 กำลังผลิตของสถานี Apy ในวันที่ 15/3/2018 จากการทำนายโดยใช้ Artificial Neural Network เปรียบเทียบกับค่าจริงที่ได้จากการวัด	63
รูปที่ 4-22 กราฟค่า performance เปรียบเทียบกันระหว่าง ชุดข้อมูลที่ใช้ในการ train แบบจำลองของสถานี Apy	63
รูปที่ 4-23 กราฟ regression เปรียบเทียบกันระหว่าง ชุดข้อมูลที่ใช้ในการ train แบบจำลองของสถานี Apy	64
รูปที่ 4-24 รูปภาพกำลังผลิตของวันที่ 29/10/2019	68
รูปที่ 4-25 รูปภาพเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ใช้ PV3 คำนวณด้วย	69
รูปที่ 4-26 ค่าเฉลี่ยของ correlation ระหว่างสถานี (PV1 ถึง PV5) ที่มีต่อกันในเวลา 1 เดือน	69
รูปที่ 4-27 รูปภาพเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 กรณีที่ไม่ใช้ PV3 ในการคำนวณ	69
รูปที่ 4-28 ACF plot และ PACF plot ของกำลังผลิต PV2	70
รูปที่ 4-29 กำลังผลิตหลังจากทำ differencing 1 ครั้ง ของ PV2 ในวันที่ 26/10/2019	70
รูปที่ 4-30 กำลังผลิตหลังจากทำ differencing 2 ครั้ง ของ PV2 ในวันที่ 26/10/2019	70
รูปที่ 4-31 ACF plot และ PACF plot ของกำลังผลิตของ PV2 หลังจากทำ differencing 2 ครั้ง	71
รูปที่ 4-32 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการทำนายเป็นผลลัพธ์จาก Kalman filter	71

รูปที่ 4-33 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator model	72
รูปที่ 4-34 กำลังผลิตของสถานี PV2 ในวันที่ 29/10/2019 จากการทำนายโดยใช้ Artificial Neural Network เปรียบเทียบกับค่าจริงที่ได้จากการวัด	73
รูปที่ 4-35 กราฟค่า performance เปรียบเทียบกันระหว่างชุดข้อมูลที่ใช้ในการ train แบบจำลองของสถานี PV2	73
รูปที่ 4-36 กราฟ regression เปรียบเทียบกันระหว่างชุดข้อมูลที่ใช้ในการ train แบบจำลองของสถานี PV2	74
รูปที่ 4-37 ค่าเฉลี่ยของ correlation ของสถานี (PV6 ถึง PV9 และ PV2) ที่มีต่อกันในเวลา 1 เดือน	78
รูปที่ 4-38 รูปภาพเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ใช้ PV6 และ PV9 ร่วมในการคำนวณ	78
รูปที่ 4-39 รูปภาพเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ไม่ใช้ PV6 และ PV9 ร่วมในการคำนวณ	79
รูปที่ 4-40 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator model คำนวณค่าประมาณข้อมูลกำลังผลิตแบบเวลาจริง กรณีระยะทางระหว่าง 10-20 กิโลเมตร.....	79
รูปที่ 4-41 ค่าเฉลี่ยของ correlation ของสถานี (PV10 ถึง PV13 และ PV2) ที่มีต่อกันในเวลา 1 เดือน.....	80
รูปที่ 4-42 รูปภาพเปรียบเทียบค่าจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ใช้ PV13 ในการคำนวณ	81
รูปที่ 4-43 รูปภาพเปรียบเทียบค่าจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ไม่ใช้ PV13 คำนวณ.....	81
รูปที่ 4-44 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator model คำนวณค่าประมาณข้อมูลกำลังผลิตแบบเวลาจริง กรณีระยะทางเกิน 20 กิโลเมตร	82
รูปที่ 4-45 ค่า correlation ระหว่าง PV ทั้งหมดในระบบ	83



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของวิทยานิพนธ์

ในปัจจุบันปัญหาที่ทำนายของการผลิตพลังงานไฟฟ้าทั่วโลกคือการเชื่อมต่อแหล่งผลิตไฟฟ้าที่ใช้พลังงานทดแทนที่มีจำนวนเพิ่มขึ้นอย่างมากในหลายปีที่ผ่านมาซึ่งเข้ากับระบบไฟฟ้าที่มีอยู่เดิมโดยเฉพาะพลังงานทดแทนที่สามารถควบคุมได้ยาก เช่น พลังงานลมและพลังงานจากแสงอาทิตย์ เนื่องจากผู้ที่ควบคุมดูแลพลังงานในระบบ (Electrical Operator) ต้องควบคุมให้พลังงานไฟฟ้าที่ผลิตได้ในระบบพอดีกับความต้องการใช้ไฟฟ้าตลอดเวลา แต่การรักษาสมดุลระหว่างการผลิตไฟฟ้า (Supply) กับความต้องการใช้ไฟฟ้า (Demand) เป็นสิ่งที่ทำได้ยาก เพราะว่าแหล่งกำเนิดพลังงานไฟฟ้าที่เกิดจากพลังงานทดแทนหรือพลังงานธรรมชาติส่วนใหญ่มีความแปรปรวนมากทำให้ยากต่อการคาดการณ์กำลังผลิตเป็นเหตุให้ประสิทธิภาพของไฟฟ้าในระบบ (Grid) ลดลง เช่น ทำให้เกิดแรงดันกระเพื่อมในระบบ (Voltage Fluctuations) มีปัญหาทางด้านคุณภาพไฟฟ้า (Power Quality) และ เสถียรภาพของระบบ (Stability) ดังนั้นการทำนายกำลังผลิตของแหล่งกำเนิดไฟฟ้าจึงมีความสำคัญอย่างมากในการเพิ่มประสิทธิภาพสำหรับการควบคุมเพื่อทำให้เกิดความสมดุลระหว่างความสามารถในการผลิตพลังงานกับความต้องการใช้พลังงาน นอกจากนี้การทำนายยังมีความสำคัญต่อการซื้อขายไฟฟ้าในตลาดซื้อขายไฟฟ้า (Electricity Market) เป็นเหตุให้ช่วยลดค่าใช้จ่ายในด้านการผลิตไฟฟ้าได้อย่างมีประสิทธิภาพ

จากที่กล่าวมาข้างต้นเราสามารถแก้ปัญหาความไม่สมดุลในระบบไฟฟ้าที่เกิดจากความแปรปรวนของแหล่งผลิตไฟฟ้าที่ใช้พลังงานทดแทนได้โดยการติดตั้งแหล่งเก็บพลังงานสำรอง (Energy Storage) เพื่อกักเก็บพลังงานไฟฟ้าส่วนที่ผลิตเกินหรือจ่ายพลังงานไฟฟ้าชดเชยในส่วนที่ผลิตไม่เพียงพอเนื่องจากความแปรปรวนได้ โดยการควบคุมแหล่งเก็บพลังงานสำรองเหล่านี้ให้จ่ายพลังงานและเก็บพลังงานในช่วงเวลาที่เหมาะสมซึ่งจะต้องอาศัยการทำนายความสามารถของแหล่งผลิตไฟฟ้าที่ใช้พลังงานทดแทนให้ได้อย่างแม่นยำ นอกจากนี้การทำนายกำลังผลิตยังมีความสำคัญต่อการเดินเครื่องผลิตไฟฟ้าเนื่องจากการเดินเครื่องผลิตไฟฟ้าแต่ละครั้งใช้เวลานาน (ขึ้นกับชนิดเชื้อเพลิงที่ใช้ในการผลิตไฟฟ้าของโรงไฟฟ้า) [1] ดังนั้นการที่รู้ค่ากำลังผลิตในอนาคตทำให้สามารถวางแผนเดินเครื่องกำเนิดไฟฟ้าล่วงหน้าได้

วิธีในการสร้างแบบจำลองการพยากรณ์กำลังผลิตไฟฟ้ามีอยู่หลากหลายวิธีซึ่งแต่ละวิธีมีข้อดี ข้อเสียและข้อจำกัดแตกต่างกันออกไปขึ้นกับหลายปัจจัยเช่น ชนิดของข้อมูลที่นำมาสร้างแบบจำลอง รูปแบบของจุดประสงค์ในการใช้งานแบบจำลองและสภาพภูมิประเทศหรือที่ตั้งสถานีโซลาร์เซลล์ เป็นต้น ดังนั้นแบบจำลองที่ใช้ได้ผลดีกับสถานีโซลาร์เซลล์หนึ่งอาจจะไม่เหมาะสมกับอีกสถานีหนึ่ง แต่โดยรวมแล้วการเลือกใช้วิธีการสร้างแบบจำลองที่เหมาะสมเราอาจจะสามารถแบ่งได้ตามช่วงเวลาในการทำนายได้โดยช่วงเวลาในการทำนาย (Time Horizon) ตั้งแต่หน่วยนาทีจนถึงการทำนายล่วงหน้าเป็นวัน เดือน หรือ ปี หรือเลือกจากความต้องการในการนำข้อมูลที่ทำนายไปใช้เพื่อประโยชน์อะไร

จากที่กล่าวมาข้างต้นนี้เองทำให้ทราบได้ว่าการทำนายกำลังผลิตไฟฟ้าล่วงหน้ามีความสำคัญทำให้เราต้องมีวิธีการทำนายกำลังผลิตที่เหมาะสมและมีประสิทธิภาพที่สุดเช่นกันโดยในวิทยานิพนธ์ฉบับนี้จะมุ่งไปที่การทำนายกำลังผลิตไฟฟ้าของแหล่งพลังงานทดแทนที่ได้จากแสงอาทิตย์ (Photovoltaic output forecasting) และจะใช้แบบจำลองที่สอดคล้องกับการทำนายระยะสั้น (Short-term forecast) มีความละเอียดมากที่สุดในหน่วยนาที เพื่อที่จะได้เปลี่ยนแปลงตามความแปรปรวนของแหล่งกำเนิดได้ทัน

1.2 งานวิจัยที่เกี่ยวข้อง

การทำนายกำลังผลิตของโซลาร์เซลล์นั้นเราทำได้ 2 วิธีเป็นหลัก วิธีแรกคือการทำนายความเข้มของแสงอาทิตย์ที่ลงมาบนพื้นโลก (Solar forecasting) จากนั้นเมื่อได้ค่าความเข้มของแสงอาทิตย์แล้วจึงเปลี่ยนเป็นกำลังผลิตจากสมการความสัมพันธ์ระหว่างความเข้มของแสงอาทิตย์กับกำลังผลิต แต่เนื่องจากความสัมพันธ์ของความเข้มของแสงกับกำลังผลิตของโซลาร์เซลล์นั้นเป็นความสัมพันธ์แบบเส้นตรงเมื่อความเข้มของแสงมากทำให้กำลังผลิตของโซลาร์เซลล์จะมากขึ้นตามไปด้วยทำให้เราสามารถทำนายกำลังผลิตของโซลาร์เซลล์จากการใช้ข้อมูลกำลังผลิตของโซลาร์เซลล์โดยตรงมาเป็นข้อมูลทางสถิติได้เช่น (Photovoltaic output forecasting) ซึ่งเป็นรูปแบบการทำนายรูปแบบที่สอง โดยรูปแบบของการสร้างแบบจำลองเพื่อทำนายกำลังผลิตของโซลาร์เซลล์แบ่งได้เป็น 2 วิธีการใหญ่ๆ [2] ได้แก่ การใช้แบบจำลอง PV performance model และ การสร้างแบบจำลองโดยวิธี Statistical model

เมื่อกกล่าวถึงวิธีการทำนายความเข้มของแสงโดยใช้แบบจำลอง PV performance model จะใช้ข้อมูลทางสภาพอากาศมาจำลองสภาพอากาศของโลกโดยการแก้สมการการ Fluid mechanics และ Thermodynamics เรียกอีกอย่างได้ว่าแบบจำลอง NWP (numerical weather prediction) ข้อดีของวิธีนี้คือเราไม่จำเป็นต้องมีข้อมูลกำลังผลิตทางสถิติเก็บไว้วิธีนี้สามารถทำนายความเข้มของแสงได้ถ้าหากทราบตัวแปรที่จำเป็นทางสภาพอากาศ แต่วิธีนี้มีข้อจำกัดเพราะว่าไม่สามารถจำลองสภาพอากาศได้อย่างสมบูรณ์แบบทำให้มีค่าความผิดพลาดอยู่เนื่องจากหลายปัจจัยทั้งทางด้านเครื่องมือวัด การคำนวณในระหว่างกระบวนการ จำนวนจุดที่เครื่องตรวจวัดค่าพารามิเตอร์ที่ติดตั้งและระยะของพื้นที่ที่ใช้ในการคำนวณมีผลต่อความผิดพลาด ดังนั้นเมื่อค่าที่ได้จากการทำนายยังไม่แม่นยำเพียงพอจึงต้องหาทางปรับแก้ค่าที่ได้จากการทำนายโดยใช้วิธีการ post-processing method มาช่วยได้แก่ Kalman Filter โดยมีข้อดีคือ ไม่ต้องใช้ข้อมูลกำลังผลิตในอดีตในการทำงาน สามารถปรับแก้ได้ทันทีแบบเวลาจริงโดยอาศัยข้อมูลในปัจจุบันและเหมาะสมกับการทำนายที่มีช่วงเวลาในการทำนาย (Time Horizon) สั้น

Sophie Pelland [3] ได้เสนอแนวทางในการใช้ post-processing method มาช่วยในการปรับแก้ทำให้ค่าความผิดพลาดที่ได้จากการทำนายความเข้มของแสงจากโมเดล NWP ให้มีค่าลดลง โดยใช้ข้อมูลความเข้มของแสงที่เก็บจากสถานีโซลาร์เซลล์ที่เราสนใจผ่านกระบวนการของ post-processing method 2 วิธีคือ spatial averaging และ Kalman filter หลังจากการทำนายปรากฏว่าผลลัพธ์ให้ค่า root mean square error (RMSE) ลดลงเฉลี่ย 15 เปอร์เซ็นต์เมื่อเทียบกับการทำนายธรรมดาที่ไม่ใช้ post-processing method

นอกจากนี้ยังมีงานวิจัยอื่นอีกที่ใช้ Kalman filter ช่วยในการลดค่าความผิดพลาดที่เกิดจากการทำนายของโมเดลแบบ NWP เช่น Maimouna Diagne [4] ใช้ post-processing method 2 วิธีคือ Model Output Statistics (MOS) และ Kalman filter ช่วยปรับปรุงผลลัพธ์ของการทำนายความเข้มของแสงอาทิตย์ล่วงหน้า 1 ชั่วโมงโดยใช้ข้อมูลค่าความเข้มของแสงจากสถานีโซลาร์เซลล์ที่สนใจแบบเวลาจริง (Real time) และใช้การทำนายความเข้มของแสงจากแบบจำลอง Weather Research and Forecasting (WRF) ซึ่งเป็นแบบจำลอง NWP รูปแบบหนึ่ง งานวิจัยนี้ทดลอง 2 สถานีโซลาร์เซลล์ผลลัพธ์ที่ได้ปรากฏว่าค่า RMSE ของสถานีแรกลดลงจาก 37.18 เหลือ 14.94 เปอร์เซ็นต์ และ สถานีที่สองค่า RMSE ลดลงจาก 57.3 เหลือ 21.8 เปอร์เซ็นต์

งานวิจัยที่กล่าวมาข้างต้นเป็นการใช้แบบจำลองการทำนายจาก PV performance model ในส่วนของการทำนายโดยแบบจำลองที่มีพื้นฐานจากค่าทางสถิติ หรือ ใช้ Statistical model นั้นมีงานวิจัยที่ใช้ Kalman filter ช่วยปรับปรุงผลลัพธ์จากการทำนายเช่นกัน โดยวิธีที่ใช้ ได้แก่ Auto-regressive model (AR) เนื่องจากวิธี Kalman filter ต้องการสมการที่สามารถเขียนอยู่ในรูปของ state-space model ได้ในการเริ่มทำงานซึ่งวิธีการสร้างแบบจำลองแบบ Auto-regressive model (AR) เป็นการจำลองค่าชุดข้อมูลที่เกี่ยวข้องกันทางเวลากล่าวาคือค่าที่เราสนใจ ณ เวลาปัจจุบันมีความสัมพันธ์กับค่าในอดีตและสามารถเขียนความสัมพันธ์นั้นในรูปของสมการได้ดังนั้นจึงสามารถเขียนในรูปของ state-space model ได้

Ted Soubdhan [5] สร้างแบบจำลองทำนายความเข้มของแสงและกำลังผลิตของโซลาร์เซลล์ 2 วิธีได้แก่ Auto-Regressive model (AR) และ Expectation-Maximization algorithm ในการทำนายเพื่อหาค่า parameter มาใช้ในการทำงานของ Kalman Filter เพื่อให้ค่าความผิดพลาดของการทำนายน้อยลงเมื่อเทียบกับการใช้วิธี Auto-Regressive (AR) หรือ Expectation-Maximization algorithm เพียงอย่างเดียว งานวิจัยนี้ใช้การประเมินผลลัพธ์โดยค่า skill score improvement ซึ่งเป็นการเปรียบเทียบค่าทำนายที่ได้จากแบบจำลองที่เราพัฒนาอยู่กับแบบจำลองที่ใช้เพื่ออ้างอิงโดยปรากฏว่าค่า skill score improvement มีค่า 39.85 เปอร์เซ็นต์ แปลว่าแบบจำลองที่เราพัฒนาสามารถทำนายได้ดีกว่าแบบจำลองอ้างอิง 39.85 เปอร์เซ็นต์

นอกจากนี้ M. Hassanzadeh [6] ได้ทำนายโดยใช้ Kalman filter ร่วมกับวิธีทำนายแบบ Auto-Regressive Moving Average (ARMA) และ วิธีการทำนายแบบ spectral analysis ซึ่งจะหาผลลัพธ์โดยการคำนวณ LSM (least square method) ทั้งสองวิธีนี้สามารถใช้ร่วมกับ Kalman filter ได้ เมื่อเปรียบเทียบผลลัพธ์ของสองวิธีนี้เมื่อใช้ร่วมกันกับ Kalman Filter ปรากฏว่าวิธี spectral analysis ให้ผลลัพธ์ดีกว่า ARMA โดยค่า Root Mean-Square Error (RMSE) ของวิธี spectral analysis อยู่ที่ 29 เปอร์เซ็นต์ และ RMSE ของวิธี ARMA อยู่ที่ 40 เปอร์เซ็นต์ เมื่อพิจารณาที่ความละเอียดของการทำนาย 5 นาที

วิธีการทำนายแบบ statistical model อีกอย่างที่ใช้กันแพร่หลายในปัจจุบันได้แก่การใช้ Artificial Neural Network (ANN) ซึ่งได้มีงานวิจัยหลายชิ้นที่ประยุกต์ใช้วิธีนี้ในการทำนายกับ เซลล์แสงอาทิตย์ เช่น L. Alfredo Fernandez-Jimenez [7] ได้เสนอแนวทางในการทำนายการผลิตไฟฟ้าจากพลังงานแสงอาทิตย์ในระยะสั้นโดยใช้ Statistical model โดยระบบประกอบไปด้วย 3 ส่วนได้แก่ Numerical Weather Prediction (NWP) model ใช้ทำนายตัวแปรของสภาพอากาศ 2 ส่วนเพื่อนำไปใช้ในส่วนที่สามซึ่งเป็นส่วนที่ใช้ทำนายพลังงานที่ได้จากระบบโดยใช้ Artificial Neural Network ในการทำนาย โดยที่ข้อมูลที่ป้อนเข้าระบบการทำนายโดยใช้ Artificial Neural Network ได้แก่ข้อมูลพลังงานที่สามารถผลิตได้ในอดีตและข้อมูลของสภาพอากาศที่ได้จากการทำนายโดยใช้ NWP model และในงานวิจัยได้เปรียบเทียบแบบจำลองชนิดต่างๆด้วย ผลลัพธ์แสดงให้เห็นว่า ANN ให้ผลลัพธ์ของค่า RMSE แม่นยำกว่าแบบจำลองชนิดอื่น

A. Mellit [8] ได้ใช้แนวทางของ Artificial Neural Network (ANN) ในการสร้างแบบจำลองเพื่อทำนายกำลังผลิตของเซลล์แสงอาทิตย์โดยแบ่งแบบจำลองที่พัฒนาออกเป็น 2 ชนิดตามชนิดของสภาพอากาศได้แก่ วันที่สภาพอากาศแจ่มใสและวันที่ท้องฟ้ามีเมฆมาก ผลลัพธ์ที่ได้ยืนยันความแม่นยำของแบบจำลองที่ใช้ neural network ในการทำนายได้เป็นอย่างดี อีกทั้งเมื่อเปรียบเทียบกับแบบจำลองแบบ regression อย่างเดียวแล้วพบว่ามีความสามารถในการทำนายที่ดีกว่า โดยข้อดีของการใช้ ANN คือไม่ต้องใช้พารามิเตอร์ที่ซับซ้อนและจำนวนมากในการสร้างและพัฒนาโมเดล

วิทยานิพนธ์ฉบับนี้มุ่งเน้นไปที่การทำนายและติดตามกำลังผลิตโซลาร์เซลล์แบบระยะสั้นของพื้นที่หรือ solar PV ที่เราสนใจโดยใช้ชุดข้อมูลกำลังผลิตของโซลาร์เซลล์ในอดีตในการทำนาย โดยในขั้นแรกเราจะใช้วิธี Auto-Regressive Integrated Moving Average (ARIMA) ในการสร้างแบบจำลองเพื่อทำนายกำลังผลิตจากข้อมูลทางสถิติในอดีตเพื่อให้ได้พารามิเตอร์ในการใช้งานในขั้นตอนต่อไป จากนั้นจะใช้กระบวนการ post-processing method ได้แก่ Kalman filter ในการทำนายโดยใช้ข้อมูลกำลังผลิตของสถานีโซลาร์เซลล์ที่เราสนใจแบบเวลาจริง (Real time) นอกจากนี้เรายังเสนอแนวทางในการแก้ปัญหาในกรณีที่เราไม่มีข้อมูลกำลังผลิตแบบเวลาจริง (Real time) ของสถานีที่เราสนใจ โดยเราจะใช้ข้อมูลของกำลังผลิตของสถานีโซลาร์เซลล์ข้างเคียงมาสร้างแบบจำลองเพื่อประมาณค่ากำลังผลิตของสถานีโซลาร์เซลล์ที่เราสนใจเราเรียกแบบจำลองนี้ว่า Estimator model แบบจำลองนี้เมื่อใช้ร่วมกับ Kalman filter จะทำให้เราสามารถติดตามกำลังผลิตแบบเวลาจริงได้ นอกจากนี้วิธีที่เราเสนอยังสามารถ

ประมาณค่ากำลังผลิตย้อนหลังถ้าหากข้อมูลกำลังผลิตขาดหายไปได้อีกด้วย ซึ่งวิธีนี้มีประโยชน์อย่างมากต่อการเก็บข้อมูลทางสถิติเพื่อเก็บข้อมูลหรือนำข้อมูลไปวิเคราะห์การทำงานของ PV ในด้านอื่นๆได้ ในขั้นตอนของการเปรียบเทียบผลลัพธ์เราจะคำนวณค่าดัชนีความแม่นยำเปรียบเทียบผลลัพธ์จากการทำนายโดยวิธี ARIMA-Kalman ที่เราเสนอกับการทำนายแบบ Artificial Neural Network และวิเคราะห์ข้อดีข้อเสียของการใช้ Estimator model ในกระบวนการทำนาย

1.3 วัตถุประสงค์ของวิทยานิพนธ์

1. เพื่อออกแบบแบบจำลองในการทำนายและติดตามกำลังผลิตของโซลาร์เซลล์เป็นเวลา 1 วัน (short-term solar forecast) โดยมีความละเอียดของช่วงเวลาในการทำนายทุก 5 นาที
2. เพื่อออกแบบวิธีการในการทำนายกำลังผลิตของสถานีโซลาร์เซลล์ที่สนใจโดยใช้ข้อมูลกำลังผลิตที่วัดได้จากสถานีโซลาร์เซลล์แบบเวลาจริงมาใช้ในกระบวนการของ Kalman filter เพื่อเพิ่มความแม่นยำในการทำนาย
3. เพื่อออกแบบและพัฒนาโมเดลการประมาณค่ากำลังผลิตที่สร้างจากข้อมูลกำลังผลิตของสถานีโซลาร์เซลล์ที่อยู่ข้างเคียงกับสถานีที่เราสนใจเพื่อนำมาใช้กับกระบวนการ Kalman filter ในกรณีที่ไม่มีข้อมูลกำลังผลิตแบบเวลาจริงของสถานีที่สนใจ
4. เพื่อทดสอบความสามารถของวิธีการที่ได้คิดขึ้นและวิเคราะห์ถึงข้อดีข้อเสียของการใช้โมเดลการประมาณค่ากำลังผลิตในการทำงานร่วมกับกระบวนการ Kalman filter เพื่อทำนายกำลังผลิตแบบเวลาจริง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. วิธีการทำนายกำลังผลิตของโซลาร์เซลล์ในกรณีที่ใช้แบบจำลอง Auto-regressive integrated moving average (ARIMA) ร่วมกับ Kalman filter สามารถช่วยให้การทำนายกำลังผลิตมีความแม่นยำมากขึ้นเมื่อเปรียบเทียบกับการใช้แบบจำลอง ARIMA ทำนายการผลิตเพียงอย่างเดียว
2. ในกรณีที่ไม่มีค่ากำลังผลิตของสถานีที่สนใจแบบเวลาจริงสามารถใช้โมเดลประมาณค่าเพื่อประมาณค่ากำลังผลิตเพื่อนำไปใช้แทนข้อมูลแบบเวลาจริงได้และยังให้ผลลัพธ์ที่ใกล้เคียงกับการใช้ข้อมูลที่สถานีวัดแบบเวลาจริง
3. สามารถนำแนวทางการทำนายในวิทยานิพนธ์ไปต่อยอดหรือศึกษาถึงผลกระทบ ข้อดีและข้อเสีย เพื่อพัฒนาและประยุกต์ใช้ในการทำนายกำลังผลิตของโซลาร์เซลล์ได้ในอนาคต

4. ผลลัพธ์ที่ได้จากการทำนายกำลังผลิตสามารถนำไปใช้ในการวางแผนจัดการพลังงาน หรือ ช่วยให้ความคุมพลังงานที่เกิดจากแหล่งกำเนิดไฟฟ้าโซลาร์เซลล์ที่ไหลเข้าสู่ระบบให้มีความ สมดุลได้

1.5 เนื้อหาของวิทยานิพนธ์

บทที่ 1 ประกอบด้วยเนื้อหาเกี่ยวกับที่มาและความสำคัญของวิทยานิพนธ์ งานวิจัยที่ เกี่ยวข้องกับหลักการที่ใช้ในวิทยานิพนธ์ วัตถุประสงค์และประโยชน์ที่คาดว่าจะได้รับของ วิทยานิพนธ์

บทที่ 2 ประกอบไปด้วยหลักการและทฤษฎีที่ใช้ในการทำวิทยานิพนธ์ คำนียาม การวิเคราะห์ ข้อมูลที่เป็น Time series กระบวนการของ ARIMA model การระบุตัวแปรและค่าสัมประ สิริทธ์ของตัวแปรต่างๆในสมการ กระบวนการของ Kalman filter algorithm กระบวนการของ Artificial Neural Network และ ดัชนีชี้วัดความแม่นยำของการทำนาย

บทที่ 3 เนื้อหาของบทนี้จะกล่าวถึงขั้นตอนกระบวนการที่ใช้ทำนายกำลังผลิตของ PV โดยเริ่ม ตั้งแต่ขั้นเก็บข้อมูล การออกแบบ Estimator model การเทียบสัมประสิทธ์ของ ARIMA model เพื่อใช้ใน Kalman filter algorithm และ การทำนายโดยใช้ Kalman filter algorithm

บทที่ 4 ประกอบไปด้วยการทดสอบในรูปแบบต่างๆเพื่อทดสอบความสามารถของวิธีที่ เสนอแนะและทดสอบขีดความสามารถของ Estimator model รวมถึงวิเคราะห์ผลลัพธ์และ ข้อดีข้อเสียของแบบจำลอง Estimator model และแนวทางกระบวนการทำนายกำลังผลิต

บทที่ 5 เนื้อหาในบทนี้กล่าวสรุปขั้นตอนทั้งหมดในวิทยานิพนธ์และผลลัพธ์ข้อสรุปที่ได้จากการ ทำวิทยานิพนธ์ รวมถึงข้อเสนอแนะที่ควรปรับปรุงในอนาคต

บทที่ 2

แนวเหตุผลและทฤษฎีที่เกี่ยวข้อง

2.1 Time series Analysis

2.1.1 Introduction

Time series หรือ อนุกรมเวลา คือชุดข้อมูลที่มีความต่อเนื่องกันทางเวลา ซึ่งในวิทยานิพนธ์ฉบับนี้คือชุดข้อมูลของกำลังผลิตของโซลาร์เซลล์ที่มีช่องว่างของเวลาระหว่างข้อมูลเท่ากัน เราต้องการวิเคราะห์ชุดข้อมูลเพื่อหาความสัมพันธ์ที่มีต่อกันระหว่างกำลังผลิตในแต่ละช่วงเวลา ดังนั้นเราจึงใช้โมเดลทางคณิตศาสตร์มาช่วยอธิบายพฤติกรรมของข้อมูลกำลังผลิต ถ้าโมเดลสามารถคำนวณหาค่าถัดไปที่แน่นอนได้เรียกว่า deterministic model แต่เนื่องจากในความเป็นจริงมีตัวแปรอื่นๆอีกมากที่เกี่ยวข้องทำให้ไม่สามารถคำนวณหาค่าถัดไปอย่างแม่นยำได้จึงใช้ความน่าจะเป็นมาช่วย เราจะเรียกโมเดลนี้ว่า stochastic model

Stochastic model สามารถแบ่งเป็น 2 กลุ่มใหญ่ได้คือ stationary stochastic model และ non-stationary stochastic model ซึ่ง stationary model เป็นโมเดลที่ได้รับความสนใจและใช้เป็นสมมติฐานในการจำลองโมเดลทางคณิตศาสตร์หลายอย่างที่เกี่ยวกับความน่าจะเป็น โดยมีลักษณะเด่นเรียกว่า statistical equilibrium คือ มีค่า mean และ variance คงที่ไม่เปลี่ยนแปลงไปตามเวลา ตัวอย่างของ stationary model ที่เกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้ได้แก่ Auto Regressive (AR), Moving Average (MA) และ Auto Regressive Moving Average (ARMA) ในส่วนของ non-stationary model ได้แก่ Auto Regressive Integrated Moving Average (ARIMA)

2.1.2 Autocorrelation Function

จากที่ได้กล่าวไปข้างต้นว่าความเป็น stationary คือสมมติฐานเบื้องต้นของการพัฒนาโมเดลแบบอนุกรมเวลา ซึ่งตามปกติแล้วความเป็น stationary สามารถอธิบายได้จาก mean, variance และ autocorrelation function โดยหากชุดข้อมูลมีความเป็น stationary สมมติให้ชุดข้อมูลมีค่า $z_1, z_2, z_3, \dots, z_N$ และมีจำนวน N ค่าทั้ง 3 สามารถคำนวณได้ดังสมการที่ (2.1), (2.2) และ (2.3) อีกทั้งการคำนวณ autocorrelation function ยังมีความจำเป็นต่อการวิเคราะห์เพื่อหาจำนวนของพารามิเตอร์ในแบบจำลอง ARIMA ที่จะกล่าวต่อไปในหัวข้อที่ 2.1.6 Model Identification & Model Estimation รวมถึงความเหมาะสมของแบบจำลองจากค่า Residual หลังจาก fit แบบจำลอง ARIMA ในหัวข้อ 2.1.7 Model Diagnostic Checking

Mean

ค่าเฉลี่ยของชุดข้อมูลสามารถคำนวณได้จากการนำข้อมูลทุกค่ามารวมกันแล้วหารด้วยจำนวนของข้อมูลที่มีทั้งหมดดังสมการที่ (2.1)

$$\bar{z} = \frac{1}{N} \sum_{t=1}^N z_t \quad (2.1)$$

Variance

เมื่อข้อมูลเป็นข้อมูลแบบอนุกรมเวลาที่มี time step เท่ากัน ค่า Variance สามารถคำนวณได้ดังสมการที่ (2.2)

$$\sigma_z^2 = \frac{1}{N} \sum_{t=1}^N (z_t - \bar{z})^2 \quad (2.2)$$

Autocorrelation Function

Autocorrelation Function เกิดจากการนำค่า Autocorrelation coefficient (ρ_k) มา plot เทียบกับค่าล่าหลังของลำดับข้อมูล (lag k) ค่าที่ plot ลงในกราฟจะไม่มีหน่วยซึ่งเป็นค่า correlation coefficient ระหว่างข้อมูลที่ตำแหน่ง t และข้อมูลที่ตำแหน่ง $t+k$ ซึ่งมีความสัมพันธ์กับค่า Auto covariance coefficient (γ_k) ดังสมการที่ (2.3) โดยสำหรับสมมติฐานที่ว่าชุดข้อมูลที่น่าสนใจมีความ stationary จะทำให้ค่า γ_0 มีค่าเท่ากับ Variance (σ_z^2) ในสมการที่ (2.2)

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (2.3)$$

บนสมมติฐานของความ stationary ของชุดข้อมูลจะสามารถสรุปได้ว่าค่า joint probability distribution $p(z_{t_1}, z_{t_2})$ มีค่าเท่ากันทุกเวลา t_1 และ t_2 เมื่อ t_1 และ t_2 มีระยะเวลายาวห่างกัน k เท่ากัน ทำให้ค่า covariance ระหว่าง z_t และ z_{t+k} มีค่าเท่ากันตลอดเวลาถ้าหากระยะห่างทางเวลาของข้อมูลมีค่าเท่ากับ k ดังนั้นเราสามารถคำนวณค่า Auto covariance coefficient (γ_k) ได้ดังสมการที่ (2.4) เมื่อแทนค่า k ด้วยตัวเลข lag ต่างๆ จากนั้นนำค่า γ_k แทนในสมการที่ (2.3) จะได้ค่า Autocorrelation coefficient ดังสมการที่ (2.5)

$$\gamma_k = \text{cov}[z_t, z_{t+k}] = E[(z_t - \mu)(z_{t+k} - \mu)] \quad (2.4)$$

$$\rho_k = \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sqrt{E[(z_t - \mu)^2]E[(z_{t+k} - \mu)^2]}} = \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sigma_z^2} \quad (2.5)$$

2.1.3 Linear Stationary Model

Backward shift operator

Backward shift operator แทนด้วยสัญลักษณ์ B ใช้เพื่อลดรูปหรือเปลี่ยนรูปในการเขียนสมการของชุดข้อมูล มีนิยามคือ $Bz_t = z_{t-1}$ หรือหมายความว่าค่าล่าหลังไป 1 ตำแหน่งนั่นเอง ดังนั้นสามารถเขียนอยู่ในรูปทั่วไปได้ว่า $B^m z_t = z_{t-m}$ นอกจากนี้ยังมีอีกรูปแบบหนึ่งเรียกว่า backward difference operator ซึ่งนิยามว่า $\nabla z_t = z_t - z_{t-1}$ ซึ่งสามารถเขียนในเทอมของ B ได้ว่า $\nabla z_t = z_t - z_{t-1} = (1-B)z_t$

Auto Regressive Model

Auto Regressive (AR) Model เป็น stochastic model ที่จะคำนวณค่าปัจจุบันจากผลรวมเชิงเส้นของข้อมูลในอดีตและค่ารบกวน โดยกำหนดให้ค่า $z_t, z_{t-1}, z_{t-2}, \dots$ ตรงกับเวลา $t, t-1, t-2, \dots$ ตามลำดับและให้ $\bar{z}_t = z_t - \mu$ สามารถเขียนอยู่ในรูปของสมการได้ดังสมการที่ (2.6) และสามารถเขียนให้อยู่ในเทอมของ backward shift operator (B) ได้ดังสมการที่ (2.7)

$$\bar{z}_t = \phi_1 \bar{z}_{t-1} + \phi_2 \bar{z}_{t-2} + \dots + \phi_p \bar{z}_{t-p} + a_t \quad (2.6)$$

โดยที่ ϕ_p คือสัมประสิทธิ์ของค่าข้อมูลในอดีตที่ล่าหลังไป p ตำแหน่ง

a_t คือ white noise process ที่มีค่าเฉลี่ยเป็น 0 และมี variance คงที่

$$\phi(B) \bar{z}_t = a_t \quad (2.7)$$

โดยที่ $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$

โมเดลมีจำนวน parameter ทั้งหมด $p+2$ ค่า $(\mu, \phi_1, \phi_2, \dots, \phi_p, \sigma_a^2)$ โดยที่ σ_a^2 คือค่า variance ของ white noise process (a_t)

Moving Average Model

Moving Average (MA) Model จะเป็นโมเดลเขียนอยู่ในเทอมของค่าล่าหลังของ a_t และมีจำนวนของตำแหน่งล่าหลังแทนที่ด้วย q สามารถเขียนอยู่ในรูปของสมการได้ดังสมการที่ (2.8) และสามารถเขียนให้อยู่ในเทอมของ backward shift operator (B) ได้ดังสมการที่ (2.9)

$$\bar{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} \quad (2.8)$$

โดยที่ θ_q คือสัมประสิทธิ์ของค่า a_t ในอดีตที่ล่าหลังไป q ตำแหน่ง

a_t คือ white noise process ที่มีค่าเฉลี่ยเป็น 0 และมี variance คงที่

$$\bar{z}_t = \theta(B)a_t \quad (2.9)$$

โดยที่ $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$

โมเดลมีจำนวน parameter ทั้งหมด $q+2$ ค่า ($\mu, \theta_1, \theta_2, \dots, \theta_q, \sigma_a^2$) โดยที่ σ_a^2 คือค่า variance ของ white noise process (a_t)

Auto Regressive Moving Average Model

เพื่อเพิ่มความยืดหยุ่นให้โมเดลให้สามารถใช้กับชุดข้อมูลที่หลากหลายได้เราจึงรวมโมเดลที่กล่าวมาข้างต้น Auto Regressive Model และ Moving Average Model เข้าด้วยกันเรียกว่า Auto Regressive Moving Average (ARMA) Model สามารถเขียนในรูปของสมการได้ดังสมการที่ (2.10) และสามารถเขียนให้อยู่ในเทอมของ backward shift operator (B) ได้ดังสมการที่ (2.11)

$$\bar{z}_t = \phi_1 \bar{z}_{t-1} + \phi_2 \bar{z}_{t-2} + \dots + \phi_p \bar{z}_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} \quad (2.10)$$

$$\phi(B)\bar{z}_t = \theta(B)a_t \quad (2.11)$$

โมเดลมีจำนวน parameter ทั้งหมด $p+q+2$ ค่า ($\mu, \phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q, \sigma_a^2$) โดยที่ σ_a^2 คือค่า variance ของ white noise process (a_t)

2.1.4 Linear Non-Stationary Model

Auto Regressive Integrated Moving Average Model

ตามปกติแล้วชุดข้อมูล Time series ที่พบส่วนใหญ่จะอยู่ในรูปแบบของ nonstationary time series มากกว่า stationary time series ดังนั้นก่อนหน้าที่เราจะใช้ ARIMA เราจะต้องทำให้

ชุดข้อมูลเหล่านั้นเป็นชุดข้อมูล stationary เสียก่อนโดยการทำให้ differencing ซึ่งค่าพารามิเตอร์ที่เกี่ยวข้องกับการทำให้ differencing คือค่า d การทำให้ differencing คือการที่เรา plot ผลต่างของค่าปัจจุบันกับค่าก่อนหน้า ดังนั้นจำนวน sample จะลดลง 1 ทุกครั้งที่ทำให้ differencing ตามตัวอย่างต่อไป นี้ โดยกำหนดให้ w_t แทนชุดข้อมูลใหม่หลังการทำให้ differencing

$$d = 0, w_t = z_t$$

$$d = 1, w_t = z_t - z_{t-1}$$

$$d = 2, w_t = (z_t - z_{t-1}) - (z_{t-1} - z_{t-2}) = z_t - 2z_{t-1} + z_{t-2}$$

โดยเราจะสังเกตได้ว่าเมื่อ $d = 2$ จะไม่ใช่ผลต่างระหว่างค่าปัจจุบันกับค่าที่ล่าช้าไป 2 ตำแหน่ง แต่จะเป็นผลต่างระหว่าง “ผลต่างของค่าปัจจุบันกับค่าที่ล่าช้าไป 1 ตำแหน่ง” กับ “ผลต่างของค่าที่ล่าช้าไป 1 ตำแหน่งกับค่าที่ล่าช้าไป 2 ตำแหน่ง” ตามปกติแล้วเราจะทำให้ differencing แค่ 1 ถึง 2 ครั้งก็เพียงพอแล้วและอาจจะใช้ function log ช่วยในการควบคุมให้ค่าของชุดข้อมูลน้อยลงเพราะว่าเราต้องระวังว่าจะทำให้ differencing มากเกินความจำเป็นจนทำให้การวิเคราะห์ผิดพลาด

Auto Regressive Integrated Moving Average (ARIMA) เป็นโมเดลที่เกิดจากโมเดล Auto Regressive Moving Average (ARMA) และเพิ่ม parameter d เข้ามาในโมเดลและเขียนย่อว่า ARIMA (p, d, q) โดยที่ p คือจำนวน order ของ Auto Regressive และ q เป็นจำนวน order ของ Moving Average ซึ่งขั้นตอนต่อไปเราต้องหาค่า parameter เหล่านี้เพื่อกำหนดรูปแบบของโมเดลก่อนที่จะประมาณค่าสัมประสิทธิ์เพื่อจำลองโมเดลของชุดข้อมูลซึ่งจะอธิบายใน section ถัดไป โมเดลรูปทั่วไปของ ARIMA แสดงดังสมการที่ (2.12) หรือ (2.13) โดยที่ ∇^d แทนการนิยาม differencing ตามที่ได้กล่าวไปแล้วใน section 2.1.3 ซึ่งตัวแปร ∇ หมายถึงการที่รากบางส่วนของสมการ ARIMA มีค่ารากเท่ากับ 1

$$w_t = \nabla^d z_t = \mu + \phi_1 w_{t-1} + \phi_2 w_{t-2} + \dots + \phi_p w_{t-p} - a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} \quad (2.12)$$

$$\phi(B)\nabla^d z_t = \theta(B)a_t \quad (2.13)$$

โดยที่ $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$$

$\phi(B)$ มีความเป็น stationary สอดคล้องกับที่รากของสมการ $\phi(B) = 0$ ต้องมีค่ามากกว่า unit circle และ $\theta(B)$ สามารถ invertible สอดคล้องกับที่รากของสมการ $\theta(B) = 0$ ต้องมีค่ามากกว่า unit circle เช่นเดียวกัน เมื่อ $d = 0$ จะหมายความว่าโมเดลแสดงลักษณะความเป็น stationary process

ตัวอย่าง Special cases of ARIMA

โมเดลของ ARIMA มีชื่อเรียกต่างกันตามค่า parameter ที่ต่างกันออกไปยกตัวอย่างเช่น

ARIMA (1,0,0): first-order Autoregressive model

$$z_t = \mu + \phi z_{t-1} \quad (2.14)$$

ในกรณีนี้จะสามารถสรุปได้ว่าค่าของ z_t คำนวณจากค่าของชุดข้อมูลตัวเองในอดีตรวมกับค่าคงที่ ถ้าโมเดลมีความ stationary ค่าสัมประสิทธิ์ ϕ จะมีค่าน้อยกว่า 1 เสมอ

ARIMA (0,1,0): Random walk

$$z_t = \mu + z_{t-1} \quad (2.15)$$

ในกรณีนี้รูปแบบสมการอาจเรียกได้ว่าเป็น first-order autoregressive model ที่มีค่าสัมประสิทธิ์เป็น 1 เกิดจากการที่ทำ differencing 1 ครั้งให้ข้อมูลมีความ stationary

ARIMA (1,1,0): Differenced first-order autoregressive model

$$z_t - z_{t-1} = \mu + \phi(z_{t-1} + z_{t-2}) \quad (2.16)$$

ARIMA (0,1,1): Exponential smoothing

$$z_t = z_{t-1} - \theta_1 a_{t-1} \quad (2.17)$$

ARIMA (0,2,1) or ARIMA (0,2,2): Linear exponential smoothing

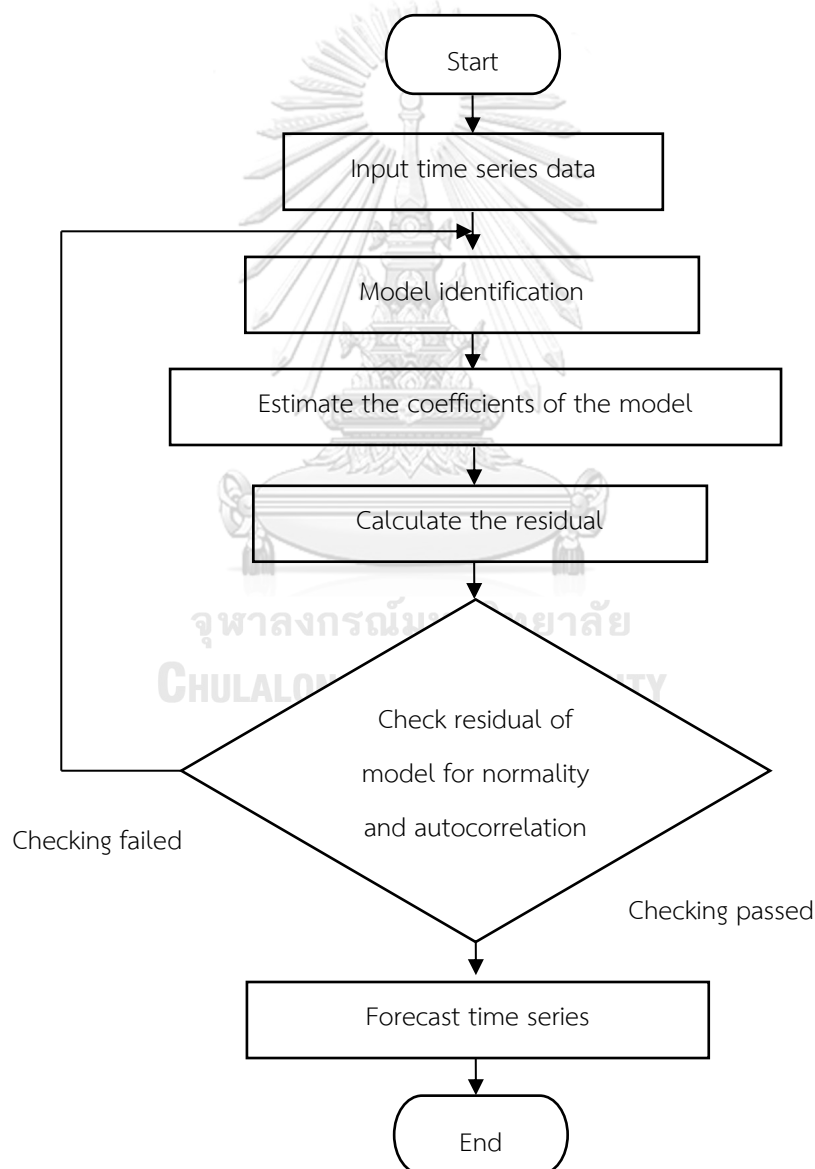
$$z_t = 2z_{t-1} - z_{t-2} - \theta_1 a_{t-1} - \theta_2 a_{t-2} \quad (2.18)$$

ARIMA (1,1,2): Damped-trend linear exponential smoothing

$$z_t = z_{t-1} + \phi(z_{t-1} + z_{t-2}) - \theta_1 a_{t-1} - \theta_2 a_{t-2} \quad (2.19)$$

2.1.5 Box & Jenkins Methodology

Box & Jenkins Methodology [9] เป็นขั้นตอนวิธีในการสร้างโมเดลของ ARIMA โดยมีขั้นตอนแยกได้เป็น 3 ขั้นตอน 1. Model Identification 2. Model Estimation 3. Model Diagnostic Checking ขั้นตอนทั้ง 3 นี้จะเรียงกันเป็นลำดับขั้นตอนและสามารถวนซ้ำซ้ำเพื่อเพิ่มประสิทธิภาพของโมเดล รูปที่ 2-1 แสดง Flowchart กระบวนการทำงานของโมเดล ARIMA ซึ่งรายละเอียดในแต่ละขั้นตอนจะอธิบายต่อไปในหัวข้อถัดไป



รูปที่ 2-1 Flowchart กระบวนการของ Box & Jenkins

2.1.6 Model Identification & Model Estimation

ในขั้นต้นชุดข้อมูล time series ส่วนใหญ่มีลักษณะที่ไม่เป็น stationary ดังนั้นต้องใช้การ differencing เพื่อให้ชุดข้อมูลมีความเป็น stationary ตามหลักการที่ได้กล่าวไปข้างต้นในหัวข้อ 2.1.4 เมื่อเราทำ differencing เสร็จแล้วเราควรจะได้ชุดข้อมูลที่มีค่า mean คงที่และไม่มี trend จากนั้นเราจะนำสัญญาณที่ได้ไปวิเคราะห์ว่าควรใช้โมเดล AR หรือ MA หรือใช้ทั้งสองอย่างร่วมกัน (ARMA) โดยเราจะวิเคราะห์จากการ plot ค่า Autocorrelation Coefficient (ACF) และ Partial Autocorrelation Coefficient (PACF) ค่า ACF คือ correlation coefficient ระหว่างค่าของ time series กับค่าล่าหลังของตัวเอง PACF คือค่าบางส่วนของค่า correlation coefficient ระหว่าง time series กับค่าล่าหลังของตัวเอง โดยสามารถใช้ตารางที่ 4-2 สรุปได้ว่าโมเดลของเราควรจะใช้ค่า p และ q เท่าใด

ตารางที่ 2-1 ความสัมพันธ์ของค่า p, q กับกราฟ ACF, PACF

Model	ACF	PACF
AR(p)	กราฟมีลักษณะลดลงอย่างช้าๆ	กราฟมีค่าต่ำกว่าเส้นลิมิตหลังจากล่า หลังมา p ตำแหน่ง
MA(q)	กราฟมีค่าต่ำกว่าเส้นลิมิตหลังจาก ล่าหลังมา q ตำแหน่ง	กราฟมีลักษณะลดลงอย่างช้าๆ
ARMA(p, q)	กราฟมีลักษณะลดลงอย่างช้าๆ	กราฟมีลักษณะลดลงอย่างช้าๆ

2.1.7 Model Diagnostic Checking

หลังจากที่เราสร้างโมเดลและหาค่าประมาณของ parameter ในโมเดลครบแล้ว เราจะตรวจสอบความเหมาะสมของโมเดลเรียกว่า goodness of fit การที่เราตรวจสอบความเหมาะสมของโมเดลสามารถช่วยในการพัฒนาโมเดลให้ดียิ่งขึ้นหรือปรับเปลี่ยนค่า parameter เช่นเพิ่มเทอมของ AR หรือ MA เป็นต้น วิธีการในการตรวจสอบความเหมาะสมของโมเดลได้แก่ 1. การทดสอบโมเดลหลากหลายแบบและเพิ่มลดค่า parameter 2. ตรวจสอบ Residual ของโมเดลว่าขัดกับสมมติฐานหรือไม่ ตรวจสอบความเป็น normality และตรวจสอบ Residual autocorrelation

Check Residual for Normality

สมมติฐานพื้นฐานของโมเดลแบบ time series คือมีความเป็น Gaussian distribution หลังจากที่เรา fit โมเดลเราสามารถคำนวณค่า Residual และตรวจสอบความเป็น Normality ถ้าสมมติฐานที่ว่า Gaussian distribution เป็นจริงเมื่อ plot Residual แล้วควรจะดูคล้าย normal distribution ตัวอย่างกราฟที่สามารถพิจารณาความเป็น normality ได้แก่ Histogram, Box plot, Quantile-Quantile plot และ Kernel density estimate

Check Residual for autocorrelation

สมมติฐานของโมเดล time series ส่วนของ noise หรือ innovation process จะสมมติว่ามีความ uncorrelated ดังนั้นหลังจากที่เรา fit โมเดลเรียบร้อยแล้ว เราสามารถตรวจสอบค่า autocorrelation ของ Residual ได้โดยการ plot sample ACF และ PACF ถ้ากราฟทั้งสองมีกราฟใดที่แสดงว่าค่า autocorrelation มีนัยสำคัญแสดงว่าเราควรปรับจำนวน parameter ในโมเดล ยกตัวอย่างเช่น เพิ่มเทอม AR หรือ MA ในโมเดล หรือใช้แบบทดสอบ Ljung-Box Q-test ซึ่งการทดสอบนี้จะทดสอบค่า autocorrelation ได้หลาย lag ในคราวเดียวโดยกำหนดค่า lag m แบบทดสอบจะทดสอบค่า autocorrelation ว่า uncorrelated จริงหรือไม่โดยไล่ตั้งแต่ lag 1 จนถึง lag m ในการตรวจสอบ ACF plot จะมีค่า standard error ที่เป็นค่าขีดจำกัดที่บ่งบอกว่าค่า correlation ของ Lag ต่างๆมีค่าเกินหรือไม่สามารถคำนวณดังนี้
$$SE_\rho = \sqrt{(1 + 2 \sum_{i=1}^{h-1} \hat{\rho}_i^2) / N}$$
 โดยค่า N คือจำนวนของข้อมูลทั้งหมด h คือลำดับ lag และ $\hat{\rho}_i$ คือค่า autocorrelation ที่ลำดับ lag i จากนั้นเส้นขีดจำกัดที่ลากขนานแกน x ใน ACF Plot จะเท่ากับ บวกลบสองเท่าของค่า SE_ρ

2.2 Kalman Filter

ตามปกติแล้วการวัดค่าโดยใช้เครื่องมือจะมีความคลาดเคลื่อนของค่าที่วัดได้ เราอาจจะวัดได้ค่าที่มากกว่า หรือ น้อยกว่าค่าที่แท้จริง ถ้าเราอยากได้ค่าที่แท้จริงเราอาจจะต้องวัดหลายครั้งแล้วใช้การหาค่าเฉลี่ยในการหาค่าที่แท้จริงที่ถูกต้อง แต่ Kalman filter คือกระบวนการวนซ้ำทางคณิตศาสตร์ที่ใช้สมการทางคณิตศาสตร์และชุดข้อมูลที่ได้จากการวัดเพื่อประมาณค่าที่แท้จริงอย่างรวดเร็วโดยที่ชุดข้อมูลที่ได้จากการวัดนั้นต้องมีความไม่แน่นอนหรือความคลาดเคลื่อนอยู่ภายในชุดข้อมูล กระบวนการทำงานสามารถแบ่งได้เป็น 2 ขั้นตอนหลักคือ State forecast และ Filtered state

2.2.1 State Forecast

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{A}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}\hat{\mathbf{u}}_t \quad (2.20)$$

$$\mathbf{P}_{t|t-1} = \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T \quad (2.21)$$

$$\mathbf{y}_t = \mathbf{C}\hat{\mathbf{x}}_{t|t-1} + \mathbf{D}\hat{\mathbf{v}}_t \quad (2.22)$$

โดยที่

$\hat{\mathbf{x}}_{t|t-1}$ คือ ตัวแปร state ที่เวลา t ที่ถูกทำนายโดยใช้ข้อมูลตั้งแต่เวลา 1, 2, 3, ..., $t-1$

$\hat{\mathbf{x}}_{t-1|t-1}$ คือ เป็นตัวแทนของตัวแปร state ในอดีตตั้งแต่เวลา 1, 2, 3, ..., $t-1$

$\hat{\mathbf{u}}_t$ คือ ตัวแปรควบคุมของสมการ

$\mathbf{P}_{t|t-1}$ คือ error covariance matrix ที่ถูกทำนาย ที่เวลา t โดยใช้ข้อมูลตั้งแต่เวลา 1, 2, 3, ..., $t-1$

$\mathbf{P}_{t-1|t-1}$ คือ error covariance matrix ในอดีตตั้งแต่เวลา 1, 2, 3, ..., $t-1$

\mathbf{A} คือ transition matrix ที่อธิบายการเปลี่ยน state จากเวลา $t-1$ ไปยัง t

\mathbf{B} คือ transition matrix ของตัวแปรควบคุมที่บ่งบอกความไม่แน่นอนของระบบ

y_t คือ Observation forecast ตัวแปรแทน ค่าสังเกตการณ์ที่ควรเกิดขึ้นเมื่อตัวแปร state มีค่าเท่ากับ $\hat{\mathbf{x}}_{t|t-1}$

C คือ transition matrix ที่อธิบายความสัมพันธ์ระหว่างตัวแปร state กับ observation

D คือ matrix หรือค่าคงที่ขึ้นกับขนาดของตัวแปรอื่นในระบบ บ่งบอกความไม่แน่นอนของการวัด

$\hat{\mathbf{V}}_t$ คือ ตัวแปรควบคุมของสมการ

ขั้นตอน State forecast เป็นขั้นตอนทำนาย state และ error covariance ที่จุดถัดไปจากสมการ state space โดยเราจะป้อนข้อมูล $\hat{\mathbf{x}}_{t-1|t-1}$ และ $\mathbf{P}_{t-1|t-1}$ ให้เป็นค่าเริ่มต้นของการทำงาน จากนั้นกระบวนการ Kalman filter จะเริ่มทำงานที่สมการ (2.20) โดยจะทำนาย $\hat{\mathbf{x}}_{t|t-1}$ โดยใช้ข้อมูลที่มีอยู่ในอดีตจนถึงช่วงเวลา $t-1$ จากนั้นขั้นตอนต่อไปจะคำนวณ $\mathbf{P}_{t|t-1}$ ตามสมการ (2.21) โดยใช้ข้อมูล $\mathbf{P}_{t-1|t-1}$ ผ่าน transition matrix \mathbf{A} และ transition matrix \mathbf{B} เมื่อคำนวณค่า $\hat{\mathbf{x}}_{t|t-1}$ และ $\mathbf{P}_{t|t-1}$ เรียบร้อยแล้วจะเข้าสู่ขั้นตอน Filtered states

2.2.2 Filtered States

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{C}^T (\mathbf{C} \mathbf{P}_{t|t-1} \mathbf{C}^T + \mathbf{D} \mathbf{D}^T)^{-1} \quad (2.23)$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (y_t - \mathbf{C} \hat{\mathbf{x}}_{t|t-1}) \quad (2.24)$$

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{C}) \mathbf{P}_{t|t-1} \quad (2.25)$$

โดยที่

$\hat{\mathbf{x}}_{t|t}$ คือ ค่าทำนายซึ่งผ่านการปรับแก้จาก Kalman gain มาแล้วเรียกว่า Filtered state

y_t คือ ค่าชุดข้อมูลที่ได้จากการวัดค่าจริงเรียกว่า observation measurement

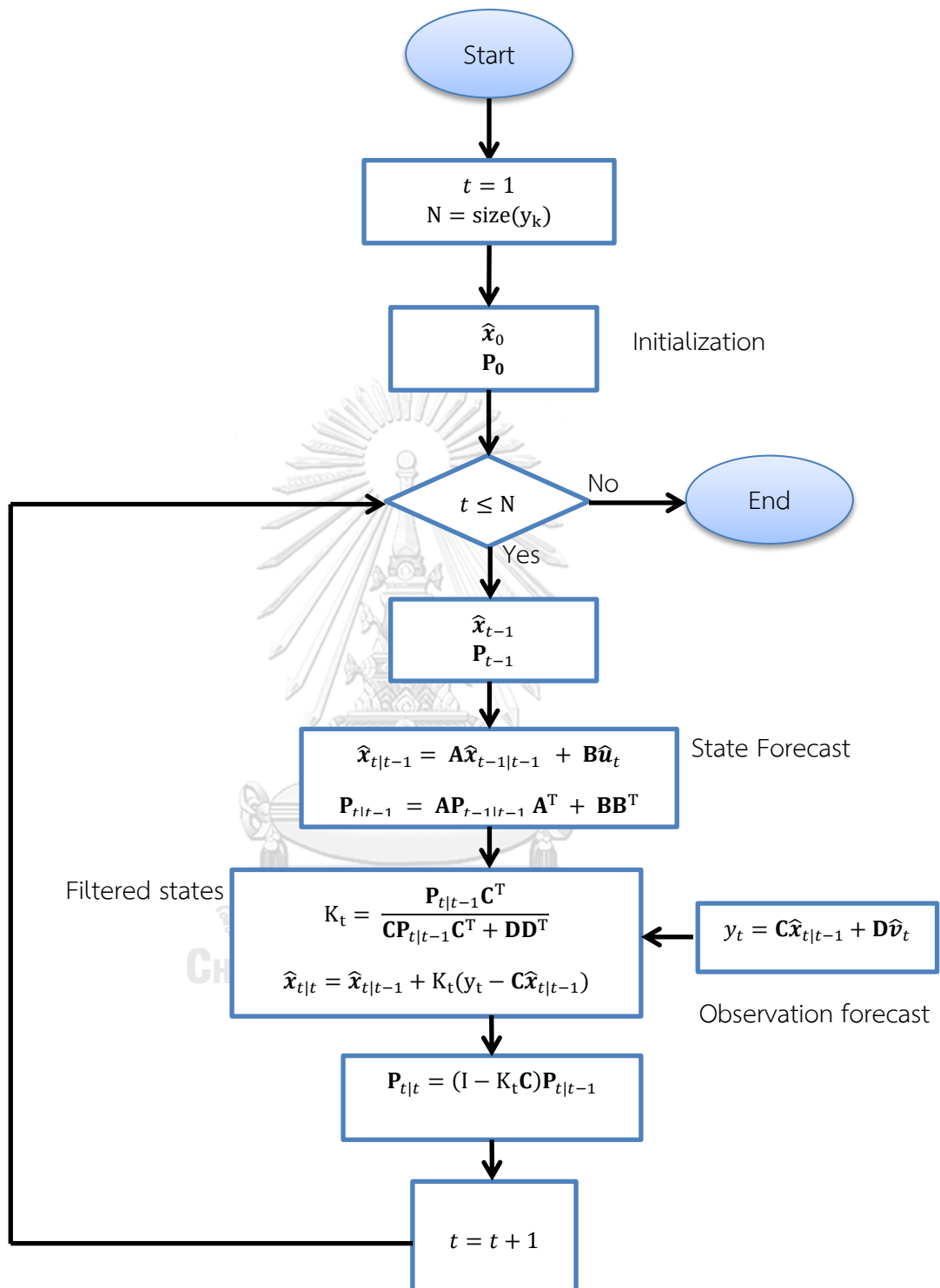
K_t	คือ ค่า Kalman gain
I	คือ Identity matrix ที่มีขนาดหลักและแถวเท่ากับจำนวนตัวแปร state
$P_{t t}$	คือ ค่า error covariance ที่เวลา t ที่ถูกปรับโดย Kalman gain

จากขั้นตอนของ state forecast ที่ผ่านมาเราจะได้อ่านค่าทำนายของ $\hat{\mathbf{x}}_{t-1|t-1}$ ที่เวลาถัดไป ซึ่งก็คือ $\hat{\mathbf{x}}_{t|t-1}$ ในขั้นตอนของ Filtered states จะนำค่าที่ทำนายได้จาก state forecast มาปรับปรุงเพื่อให้แม่นยำขึ้นซึ่งก็คือ $\hat{\mathbf{x}}_{t|t}$ โดยใช้สมการที่ (2.24) ซึ่งจะมีตัวแปร K_t หรือค่า Kalman gain ที่สามารถคำนวณได้จากสมการที่ (2.23) และค่า y_t หรือค่าที่ได้จากการวัดจริง

ในสมการที่ (2.24) เทอม $y_t - C\hat{\mathbf{x}}_{t|t-1}$ เป็นการนำค่าที่ได้จากการวัดจริงมาลบกับค่า observation ที่ได้จากการทำนายเพื่อพิจารณาว่าส่วนต่างของเทอมนี้ควรจะมีผลต่อการทำนายมากหรือน้อย โดยวิเคราะห์ได้จากค่า Kalman gain ที่คุณอยู่กับเทอมนี้ ค่า Kalman gain จะเปรียบเทียบค่าความไม่แน่นอนของการคำนวณกับค่าความไม่แน่นอนของการวัด ถ้าหากค่าความไม่แน่นอนของการวัดมีค่ามากกว่า ค่า Kalman gain จะมีค่าน้อยส่งผลให้เทอม $y_t - C\hat{\mathbf{x}}_{t|t-1}$ ส่งผลกระทบบกับค่า $\hat{\mathbf{x}}_{t|t-1}$ ไม่มากนักเพราะเราเชื่อค่าที่ได้จากการคำนวณสมการของระบบมากกว่า แต่ถ้าหากค่าความไม่แน่นอนของการคำนวณของระบบมีค่ามากกว่า ค่า Kalman gain จะเข้าใกล้ 1 ทำให้เทอม $y_t - C\hat{\mathbf{x}}_{t|t-1}$ มีผลกระทบบกับค่า $\hat{\mathbf{x}}_{t|t-1}$ มากส่งผลให้ค่าที่ได้จากการทำนายของระบบเปลี่ยนไปมาก

หลังจากคำนวณสมการที่ (2.24) แล้วจะได้ค่า $\hat{\mathbf{x}}_{t|t}$ ซึ่งเป็นค่าผลลัพธ์จากการทำนายที่ถูกต้อง หลังจากนั้นจะคำนวณสมการที่ (2.25) เพื่อหาค่า $P_{t|t}$ ซึ่งคือค่า error covariance ที่ถูกปรับแก้โดย Kalman gain และทั้งสองค่านี้จะนำไปใช้ในการคำนวณในรอบถัดไปตามลำดับ

จากสมการที่กล่าวมาข้างต้นเราสามารถเขียน Flow chart แสดงลำดับการทำงานของ Kalman filter algorithm ตามรูปที่ 2-2 โดยการทำนายจะทำนายล่วงหน้า 1 step เสมอและวนซ้ำไปเรื่อยๆหรือจนกว่าจะครบตามจำนวนค่า observation ที่ได้จากการวัดแบบเวลาจริง y_t

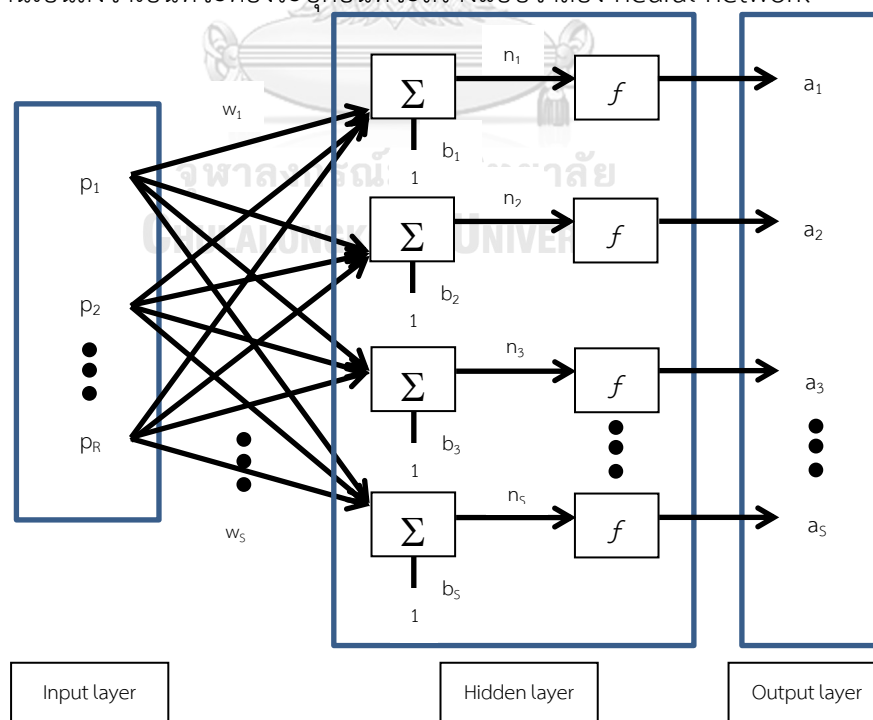


รูปที่ 2-2 Flow chart แสดงการทำงานของ Kalman filter

2.3 Artificial Neural Network (ANN)

ในงานวิจัยนี้ใช้วิธีการทำนายที่มี Artificial Neural Network ในการเปรียบเทียบผลการทดสอบกับวิธี ARIMA-Kalman และ Estimator model ที่จะนำเสนอในบทที่ 3 โดยในส่วนของบทที่ 2 นี้จะอธิบายภาพรวมของการใช้ ANN ขั้นตอนวิธีการสร้างแบบจำลองและการทำนายโดยใช้แบบจำลองที่สร้างขึ้นเป็นดังนี้

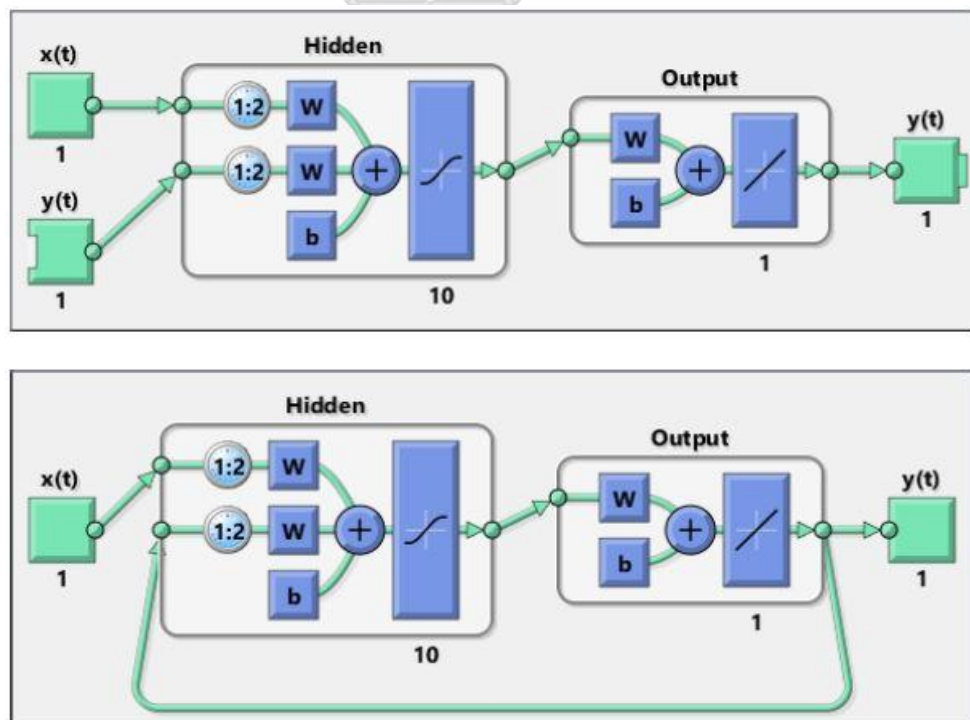
Neural Network อาจสามารถจำแนกตามชนิดของข้อมูลขาเข้า (input) ได้เป็น Static network และ dynamic network โดยชนิดของระบบจะส่งผลต่อโครงสร้างในการสร้างแบบจำลอง Static network หรือ feedforward network เป็นรูปแบบของ neural network ที่ใช้กันอย่างแพร่หลายที่สุด มีโครงสร้างทั่วไปดังรูปที่ 2-3 โดยรูปแบบของข้อมูลขาเข้าจะไม่ขึ้นกับเวลาและไม่มี feedback หรือ delays ตัวแปร p คือเวกเตอร์ข้อมูลขาเข้า ตัวแปร w คือค่าน้ำหนักของข้อมูลขาเข้าไปยัง neural แต่ละตัว ตัวแปร a คือเวกเตอร์ข้อมูลขาออก ระบบจะประกอบไปด้วย 3 ชั้นได้แก่ Input layer, Hidden layer และ Output layer ใน hidden layer สามารถมีหลาย layer ต่อกันไปได้อีกซึ่งในแต่ละ hidden layer จะประกอบไปด้วยค่าน้ำหนักและฟังก์ชันที่เชื่อมต่อกับข้อมูลขาเข้าของ Input layer ซึ่งจำนวนข้อมูลขาเข้า, ข้อมูลขาออก, จำนวน layer และ ชนิดของฟังก์ชันข้อมูลเหล่านี้เป็นสิ่งจำเป็นที่จะต้องระบุก่อนที่จะสร้างแบบจำลอง neural network



รูปที่ 2-3 โครงสร้างทั่วไปของ Artificial Neural Network

จากรูปที่ 2-3 ตัวแปร p เวกเตอร์ข้อมูลขาเข้ามีขนาด $R \times 1$ และตัวแปรน้ำหนัก w สามารถเขียนอยู่ในรูปเมทริกซ์ที่มีขนาด $S \times R$ ได้เมื่อ S คือจำนวน neuron ในระบบและ R คือจำนวนของข้อมูลขาเข้า ผลลัพธ์ของ wp จะรวมเข้ากับค่า bias ที่เกิดจากเวกเตอร์ b ที่มีขนาด $S \times 1$ จากนั้นผ่านเข้าไปยัง transfer function และได้ข้อมูลขาออกคือเวกเตอร์ a

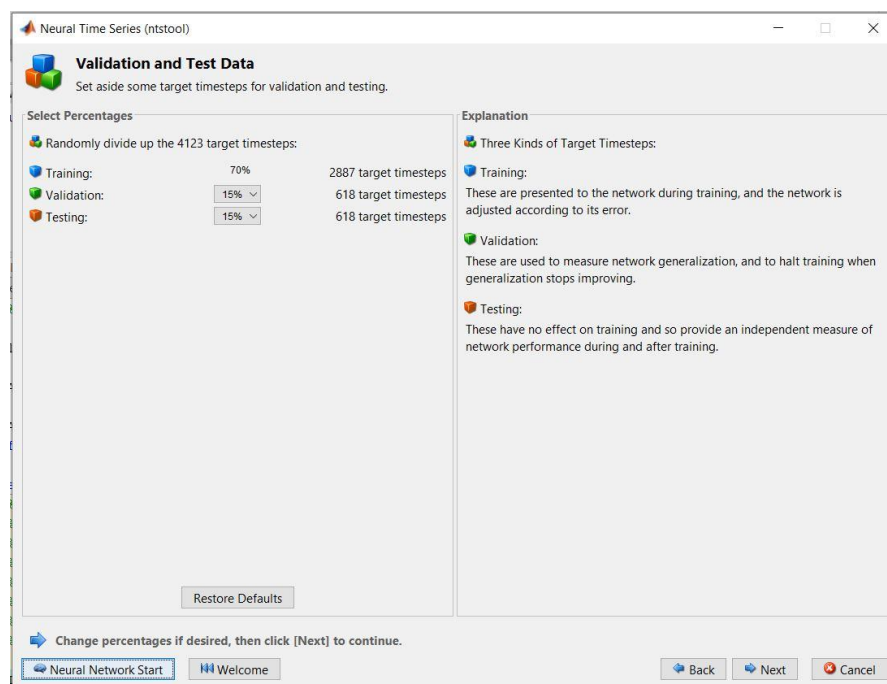
ในส่วนของ dynamic neural network จะมีโครงสร้างของระบบคล้ายกับรูปที่ 2-3 แต่ความสามารถในการคำนวณมากกว่าแบบ static network เนื่องจาก dynamic network มีหน่วยความจำสามารถพัฒนาแบบจำลองให้เรียนรู้รูปแบบและลำดับของข้อมูลได้ ในงานวิจัยนี้เราเลือกสร้างแบบจำลองแบบ dynamic network เพื่อมาเปรียบเทียบผลลัพธ์กับวิธีการที่เราได้เสนอ เนื่องจากข้อมูลของเราเป็นข้อมูลที่มีความต่อเนื่องทางเวลา dynamic network จะเหมาะสมกับข้อมูลในลักษณะนี้ โดยแบบจำลองที่ใช้มีชื่อเรียกว่า Time Series Nonlinear Auto-Regressive feedback Neural Network (NAR) รูปที่ 2-4 แสดงตัวอย่างของแบบจำลอง NAR แบบ open loop (เพื่อการ train model) และ closed loop (เพื่อการนำไปใช้ทำนาย) ในการ train แบบจำลองสามารถใช้ MATLAB toolbox ชื่อ Neural Network Time Series ในการจำลองระบบ neural network อย่างง่ายได้โดยมีขั้นตอนดังต่อไปนี้



รูปที่ 2-4 ตัวอย่าง block diagram ของ open loop (รูปบน)
และ closed loop (รูปล่าง)

Dividing data set

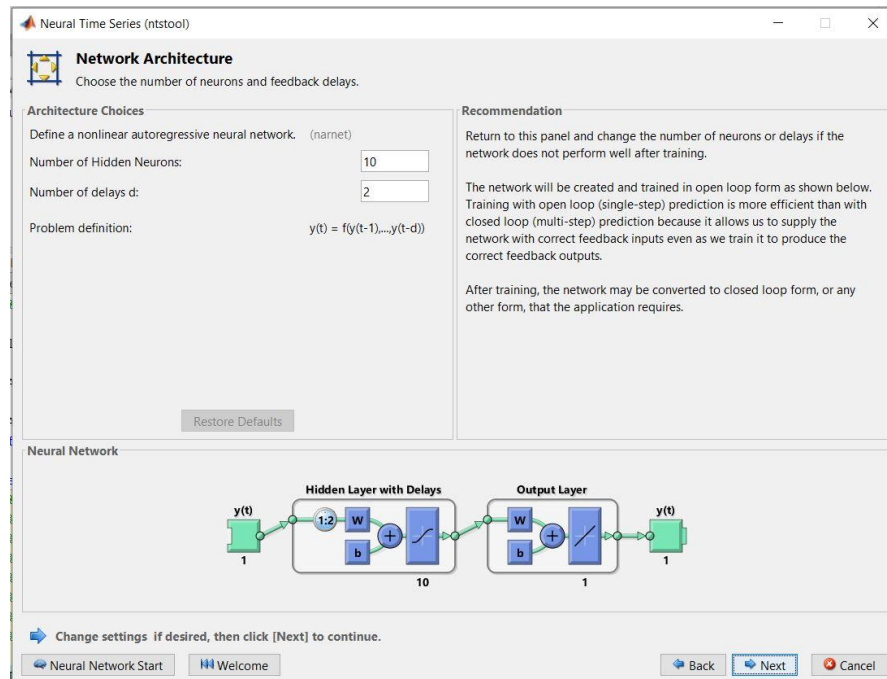
เลือก Neural Network Time Series จาก Apps ใน MATLAB จากนั้นเลือกข้อมูลที่เรามีป้อนเข้าไปใน MATLAB และแบ่งข้อมูลออกเป็น 3 ส่วนเพื่อ train ระบบได้แก่ Training set, Validation set และ Testing set ดังภาพที่ 2-5 ซึ่งรูปแบบที่นิยมใช้กันมากคือแบ่งข้อมูลออกเป็น Training set 70%, Validation set 15% และ Testing set 15% แต่สามารถปรับเปลี่ยนได้ตามที่ผู้ทดสอบต้องการเช่น Training set 80%, Validation set 10% และ Testing set 10% ขึ้นกับชนิดของข้อมูลและจำนวนของข้อมูลที่มีอยู่



รูปที่ 2-5 การแบ่งข้อมูลที่ป้อนเข้าไปเพื่อ train แบบจำลอง

Creating Neural Network

หลังจากนั้นเราระบุจำนวน neuron ใน hidden layer และจำนวนของ delays (ในกรณีที่รูปแบบของข้อมูลเป็นแบบอนุกรมเวลาและเราต้องการผลลัพธ์จากการทำนาย) จำนวน neural ที่เป็นมาตรฐานส่วนใหญ่คือ 10 ซึ่งเพียงพอต่อการสร้างระบบการแก้ปัญหาส่วนใหญ่ แต่ถ้า train แล้วได้ผลลัพธ์ที่ยังไม่เป็นที่พอใจสามารถเพิ่มจำนวน neuron ได้เพื่อเพิ่มความสามารถในการทำงานซึ่งมีข้อเสียคือถ้าเพิ่มจำนวน neuron มากเกินไปจะกระทบกับเวลาที่ใช้ในการคำนวณ ตัวแปรที่สำคัญอีกชนิดคือตัวแปร delays ใช้เพื่อบอกจำนวนของข้อมูลที่เป็นข้อมูลตั้งต้นที่จำเป็นต้องระบุเพื่อใช้ในการ train แบบจำลองแบบที่มี feedback



รูปที่ 2-6 ขั้นตอนระบุจำนวน neuron และ delays ของระบบ

Training Algorithm

ขั้นตอนถัดไปคือการเลือก training algorithm เราจะเลือกอัลกอริทึมให้เหมาะสมกับการทำงานและจำนวนข้อมูล อัลกอริทึมที่ใช้กันส่วนมากในการคำนวณของ neural network แสดงตามตารางที่ 2-2 โดยทั่วไปการที่จะบอกได้ว่าปัญหาที่เราต้องใช้ neural network เพื่อหาคำตอบต้องใช้ฟังก์ชันอะไรเพื่อแก้ปัญหาเป็นไปได้อย่างต้องใช้ประสบการณ์และการทดสอบหลายครั้ง แต่จากงานวิจัยส่วนใหญ่ที่ผ่านมาในอดีตเราสามารถสรุปได้ดังนี้

ตารางที่ 2-2 อัลกอริทึมที่ใช้ในการ train แบบจำลอง

อักษรย่อ	ชื่อเต็มของอัลกอริทึม
LM	Levenberg-Marquardt
RP	Resilient Backpropagation
SCG	Scaled Conjugate Gradient
BFG	BFGS Quasi-Newton

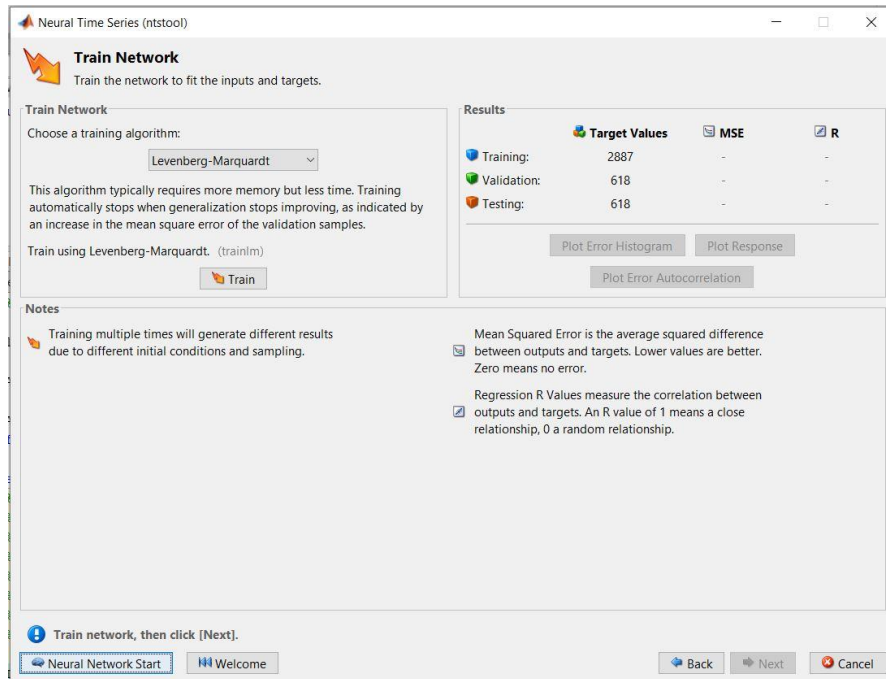
โดยทั่วไปแล้วอัลกอริทึมแบบ Levenberg-Marquardt จะใช้เวลาคำนวณที่น้อยที่สุดสำหรับปัญหาที่มีจำนวนค่าน้ำหนักในระบบประมาณหลักร้อยและได้ผลลัพธ์ที่มีความแม่นยำสูง ในหลายๆกรณีจะได้ผลลัพธ์ของค่า Mean Square Error น้อยกว่าอัลกอริทึมอื่นๆ แต่ถ้าจำนวนน้ำหนักในระบบและจำนวน neuron ในระบบเพิ่มขึ้นจะทำให้ประสิทธิภาพลดลง นอกจากนี้ อัลกอริทึมแบบ Levenberg-Marquardt จะเหมาะสมกับปัญหาแบบ function approximation มากกว่า pattern recognition

ในส่วนของอัลกอริทึมแบบ Resilient Backpropagation จะคำนวณได้เร็วที่สุดเมื่อใช้กับปัญหาแบบ pattern recognition แต่ไม่เหมาะสมกับ function approximation และมีข้อดีตรงที่ใช้หน่วยความจำน้อยกว่าอัลกอริทึมอื่น

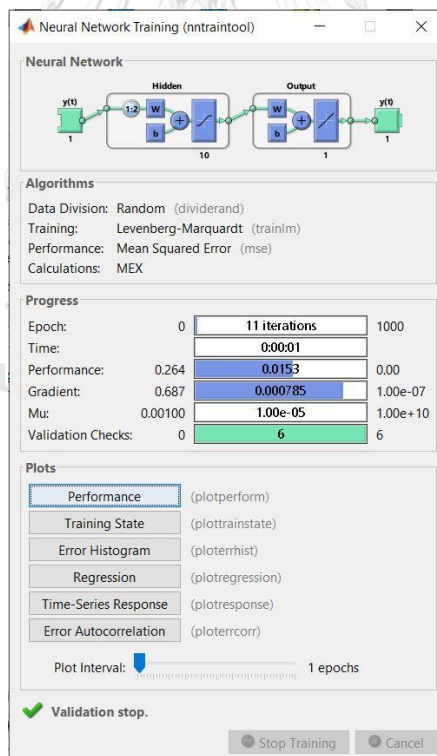
อัลกอริทึมแบบ Scaled Conjugate Gradient โดยภาพรวมแล้วมีประสิทธิภาพในการทำงานกับปัญหาหลายแบบดีเท่ากัน แต่จะมีข้อดีที่หน่วยความจำมีความจุที่มาก สามารถใช้กับระบบที่มีจำนวนน้ำหนักและจำนวน neuron ที่มากและยังมีความเร็วในการคำนวณที่ใกล้เคียงกับอัลกอริทึมแบบ Levenberg-Marquardt เมื่อใช้กับปัญหาแบบ function approximation นอกจากนี้ยังมีความเร็วในการคำนวณที่ใกล้เคียงกับอัลกอริทึมแบบ Resilient Backpropagation เมื่อใช้กับปัญหาแบบ pattern recognition อีกด้วย

อัลกอริทึมชนิดสุดท้ายที่กล่าวถึงคือ BFGS Quasi-Newton ซึ่งจะมีความสามารถและความเร็วในการคำนวณคล้ายกับอัลกอริทึมแบบ Levenberg-Marquardt ในระบบที่มีขนาดปานกลางแต่ถ้าระบบของแบบจำลองมีขนาดใหญ่ขึ้นจะให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมแบบ Levenberg-Marquardt อย่างมาก

โดยสรุปแล้วในกรณีของงานวิทยานิพนธ์ฉบับนี้ เนื่องจากข้อมูลเป็นแบบอนุกรมเวลาและเราเลือกใช้แบบจำลอง Time Series Nonlinear Auto-Regressive feedback Neural Network (NAR) ซึ่งเป็นปัญหาแบบ function approximation อีกทั้งข้อมูลที่เราเก็บมาได้ไม่ได้มีจำนวนที่มากเกินไปดังนั้น เราจึงควรเลือกใช้ อัลกอริทึมแบบ Levenberg-Marquardt เพื่อความรวดเร็วและแม่นยำในการ train แบบจำลอง โดยหลังจากขั้นตอนระบุจำนวน neuron และ delays แล้วเราเลือกอัลกอริทึมดังที่แสดงตัวอย่างในรูปที่ 2-7 ซึ่งเป็นหน้าต่างอินเทอร์เฟซที่แสดงการ train และ result นอกจากนี้ในรูปที่ 2-8 ยังแสดงถึงภาพรวมของระบบ neural network หลังการ train แบบจำลอง



รูปที่ 2-8 อินเทอร์เฟซเลือก training algorithm และการ train แบบจำลอง

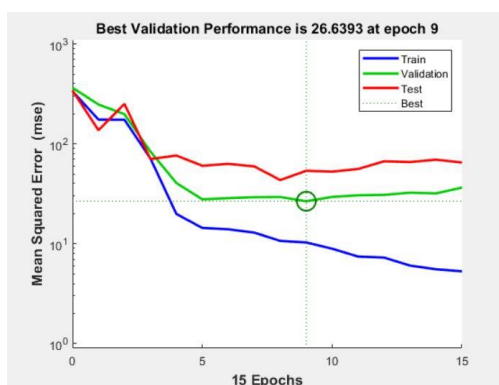


รูปที่ 2-7 อินเทอร์เฟซสรุปภาพรวมของระบบหลังการ train

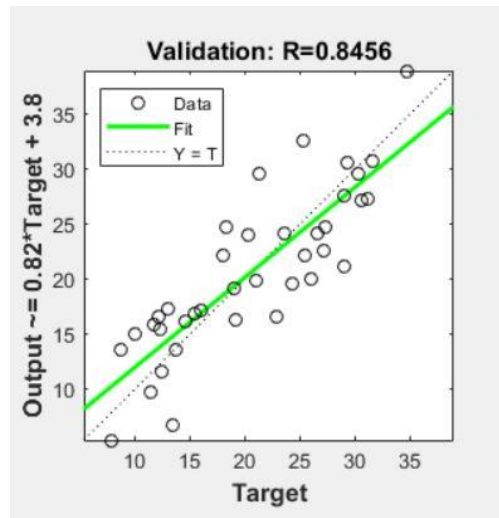
จากรูปที่ 2-8 จะเห็นได้ว่าการแสดงรายละเอียด block diagram ของระบบเพื่อช่วยต่อการเข้าใจและยังมีการระบุอัลกอริทึมที่เราใช้ในการ train แบบจำลอง นอกจากนั้น criteria ในการหยุดการ train ของโปรแกรมคือค่า Mu และ Validation check ในรูปที่ 2-8 โดยถ้าค่าอย่างใดอย่างหนึ่งต่ำกว่าขีดจำกัดที่ตั้งไว้การ train แบบจำลองจะหยุดลง โดยเมื่อค่า performance (Mean Square Error) ถึงจุดที่มีค่าน้อยที่สุดค่า Mu จะมีค่าน้อยมากถ้าหากค่า Mu น้อยกว่า $1e-5$ การ train จะหยุดลง ในส่วนของค่า Validation check คือการนับรอบการวนซ้ำของการคำนวณที่ค่า performance ไม่ได้ลดลง หรือก็คือรอบของการคำนวณรอบไหนที่ทำให้ค่า Error เพิ่มขึ้นจะนับค่า Validation check เพิ่มขึ้น 1 เมื่อค่านี้เพิ่มขึ้นจนครบ 6 ครั้งเมื่อไหร่การ train จะหยุดลง

Testing performance

หลังจากที่เรา train แบบจำลองเรียบร้อยแล้วเราจะวิเคราะห์ผลลัพธ์หลังจากการ train เพื่อที่จะตัดสินใจว่ามีความจำเป็นที่จะ train แบบจำลองใหม่อีกครั้งหรือไม่ หรือต้องการเปลี่ยนแปลงตรงจุดไหนในแบบจำลอง โดยการวิเคราะห์ performance plot และ regression plot ดังรูปที่ 2-9 และ 2-10 รูปที่ 2-9 แสดง performance plot ซึ่งแกน y แทนค่า MSE แกน x แทนจำนวนรอบในการวนซ้ำของการ train แบบจำลอง ถ้าแบบจำลองที่เรา train มีประสิทธิภาพที่ดีเส้นกราฟของ Test และ Validation จะใกล้เคียงกัน แต่ถ้าหากเส้นกราฟของ Test เพิ่มขึ้นมากกว่า Validation อย่างมากอาจแสดงให้เห็นว่าเกิดการ overfitting ขึ้นหรือแบบจำลองพยายามที่จะ fit กับ error หรือ noise มากเกินไป ส่วนกรณีของ regression plot ในรูปที่ 2-10 จะแสดงให้เห็นถึงความสัมพันธ์ระหว่างข้อมูลออกจากแบบจำลองและข้อมูลเป้าหมายของการ train ถ้าแบบจำลองมีประสิทธิภาพที่ดีความสัมพันธ์ของข้อมูลทั้งสองควรเป็นไปในทิศทางเดียวกัน โดยสามารถดูได้จากค่า R ถ้าค่า R เข้าใกล้ 1 แสดงว่าความสัมพันธ์ของข้อมูลเป็นไปในทิศทางเดียวกัน



รูปที่ 2-9 ตัวอย่างของ performance plot



รูปที่ 2-10 ตัวอย่าง regression plot ของ Validation set

Prediction

หลังจากที่ได้แบบจำลองที่ต้องการอย่างง่ายแล้วเราสามารถปรับแก้หรือเพิ่มเติมแบบจำลองได้โดยการใช้ประโยชน์จาก MATLAB ในการสร้าง script เพื่อพิมพ์โค้ดคำสั่งแก้ไขหรือเพิ่มเติมฟังก์ชันต่างในแบบจำลองได้โดยตรงโดยการกดปุ่ม Simple Script หรือ Advance Script ดังรูปที่ 2-11 ซึ่งเป็นวิธีที่รวดเร็วในการแก้ไขเปลี่ยนแปลงฟังก์ชันการคำนวณต่างๆในแบบจำลอง รวมถึงจำนวนของ neuron เพื่อใช้ในการ train แบบจำลองในรูปแบบอื่นและการทำนาย หลังจากนั้นเราจะทำนายโดยใช้คำสั่ง closed loop คำสั่งนี้เราสามารถเขียนเพิ่มเติมได้ใน script คำสั่งนี้จะเปลี่ยนแบบจำลองของเราดังรูปที่ 2-4 ที่กล่าวไปแล้วข้างต้น โดยข้อมูลขาออกจะนำมาป้อนกลับเข้าไปคำนวณใหม่ในแบบจำลองและคำตอบที่ได้จากแบบจำลองในรูปที่ 2-4 ค่า y คือค่าที่ได้จากการทำนายโดยใช้แบบจำลองที่เรา train นั้นเอง



รูปที่ 2-11 อินเทอร์เฟซการสร้าง script

2.4 การทดสอบค่าความแม่นยำของผลลัพธ์จากแบบจำลอง

การทดสอบความแม่นยำจำเป็นต่อการพัฒนาแบบจำลอง เราจะใช้การคำนวณความแม่นยำหลายแบบเพื่อวิเคราะห์แบบจำลองจากหลายแง่มุมซึ่งตามปกติแล้วการวิเคราะห์แบบจำลองที่มีลักษณะต่างกันทำได้ยากเพราะว่ามีรายละเอียดหลายอย่างที่แตกต่างกัน การวิเคราะห์แบบกราฟเราสามารถใช้ในการวาดกราฟตามช่วงเวลาในการเปรียบเทียบความแตกต่างจากค่าจริง หรือ ใช้ scatter plot เพื่อวิเคราะห์ความแม่นยำจากการทำนาย แต่ในทางการคำนวณแล้วไม่มีมาตรฐานที่แน่นอน ยืนยันหรือได้รับการยอมรับอย่างชัดเจนทำให้การเปรียบเทียบระหว่างวิธีการทำนายที่ต่างกันทำได้ยากเพราะว่าตัวแปรของแต่ละสถานที่และงานวิจัยไม่เหมือนกัน แต่งานวิจัยในอดีตหลายงานใช้การวิเคราะห์ทางสถิติในการหาค่าความแม่นยำของแบบจำลอง โดยที่ใช้ค่าผลลัพธ์ที่ได้จากการทำนาย y_{pre} ค่าที่ได้จากการวัดจริง y_{obs} และ ค่าเฉลี่ยของค่าที่ได้จากการวัดจริง y_{mean} มาคำนวณในสมการดังสมการที่ (2.26) ถึง (2.29)

2.4.1 Root Mean-Square Error (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \times \sum_{i=1}^N (y_{pre,i} - y_{obs,i})^2} \quad (2.26)$$

RMSE เป็นการคำนวณที่ให้ความสำคัญกับค่าการทำนายที่ผิดพลาดจากค่าจริงมาก ดังนั้นจะเหมาะกับแบบจำลองที่เราสามารถลดความผิดพลาดน้อยได้ถ้าจำเป็น เป็นวิธีการทดสอบความแม่นยำที่ใช้กันอย่างแพร่หลายและควรจะพัฒนาแบบจำลองให้ค่านี้มีค่าน้อยที่สุด

2.4.2 Normalized Root Mean-Square Error (nRMSE)

$$nRMSE = \frac{\sqrt{MSE}}{y_{mean}} = \frac{\sqrt{\frac{1}{N} \times \sum_{i=1}^N (y_{pre,i} - y_{obs,i})^2}}{y_{mean}} \quad (2.27)$$

2.4.3 Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{N} \times \sum_{i=1}^N \left| \frac{y_{pre,i} - y_{obs,i}}{y_{obs,i}} \right| \times 100\% \quad (2.28)$$

MAPE มีค่าใกล้เคียงกับ MAE แต่ความแตกต่างระหว่างค่าที่ได้จากการทำนายกับค่าที่ได้จากการวัดจะถูกหารด้วยค่าที่ได้จากการวัดเพื่อเทียบความห่างกับค่าที่ได้จากการวัดเป็นแบบร้อยละ แต่ว่าถ้าค่าที่ได้จากการวัดมีค่าเข้าใกล้ 0 จะไม่สามารถคำนวณได้

2.4.4 Mean Square Error (MSE)

$$MSE = \frac{1}{N} \times \sum_{i=1}^N (y_{pre,i} - y_{obs,i})^2 \quad (2.29)$$

MSE ใช้ค่ากำลังสองของความแตกต่างระหว่างค่าที่ได้จากการทำนายและค่าที่ได้จากการวัดจริงทำให้ค่าความผิดพลาดที่มีค่ามากทวีความใหญ่มากขึ้นเพื่อชี้จุดที่ผิดพลาดของแบบจำลอง MSE เป็นค่าที่ควรจะพัฒนาแบบจำลองให้ค่านี้น้อยที่สุด

2.4.5 Skill Score

นอกเหนือจากการคำนวณความแม่นยำของแบบจำลองตามสมการข้างต้นแล้วยังมีวิธีในการทดสอบความแม่นยำวิธีอื่นอีกเช่นการใช้ skill score improvement โดยการใช้แบบจำลองอ้างอิงเพื่อเป็นมาตรฐานในการทดสอบ ตัวอย่างแบบจำลองอ้างอิงที่งานวิจัยส่วนใหญ่ใช้ในการทดสอบเช่น

Persistence model นิยมใช้กันอย่างแพร่หลายเพื่อเป็นแบบจำลองอ้างอิงสำหรับการพัฒนาแบบจำลองในงานวิจัยเนื่องจากมีแนวคิดที่เรียบง่าย โดยมีสมมติฐานคือจะสมมติว่าค่าของชุดข้อมูลในอนุกรมเวลามีค่าคงที่ระหว่างช่วงเวลา t ถึง $t+h$ การใช้งานแบบจำลองอ้างอิงนี้จะใช้การเปรียบเทียบจากกราฟโดยตรงหรือใช้ skill score เพื่อวิเคราะห์ว่าแบบจำลองที่เราทดสอบมีการพัฒนาที่ดีขึ้นหรือเลวลงเมื่อเทียบกับ Persistence model

Naive Persistence model เป็นหนึ่งในแบบจำลองที่นิยมมากในหมู่ Persistence model โดยสมมติว่าข้อมูลที่ถูกทำนายในช่วงเวลาต่อไปจะมีค่าเท่ากับค่าข้อมูลสุดท้ายที่ได้จากการวัด ยกตัวอย่างเช่น สำหรับการทำนายที่ 1 ชั่วโมงข้างหน้า ข้อมูลที่เวลา 14:00 น จะมีค่าเท่ากับข้อมูลที่ 13:00 น หรือถ้าช่วงเวลาในการทำนายเป็น 1 วันจะสรุปได้ว่าค่าที่ได้จากการทำนายของวันพรุ่งนี้จะ คล้ายกันกับข้อมูลของวันนี้หรือคล้ายกันกับข้อมูลของวันที่มีการเก็บไว้ล่าสุดย้อนหลังไป 1 วัน [2]

$$P_{pre}(t + h) = P_{mea}(t) \quad (2.30)$$

โดยที่ $P_{pre}(t + h)$ คือค่ากำลังผลิตในอนาคตที่เวลา $t+h$ เมื่อ h คือระยะเวลาที่จะทำนาย

$P_{mea}(t)$ คือค่ากำลังผลิตที่เก็บวัดได้ที่เวลา t

แต่เนื่องจากข้อมูลส่วนใหญ่ไม่มีความเป็น stationary ดังนั้นชุดข้อมูลที่ใช้แบบจำลอง Naive Persistence จึงมีข้อจำกัดโดยส่วนใหญ่จะใช้กับการทำนายในระยะเวลานั้น ดังนั้นบางงานวิจัยจึงมีการเสนอแบบจำลองใหม่เรียกว่า smart persistence เช่น เมื่อใช้กับการทำนายที่เกี่ยวกับเซลล์ แสงอาทิตย์ แบบจำลองนี้จะแยกองค์ประกอบของกราฟกำลังผลิตหรือความเข้มของแสงออกเป็น ส่วนที่มีกราฟฟูปร่างเหมือนกับเวลาท้องฟ้าสดใส และ ส่วนที่มีความแปรปรวนซึ่งเกิดจากก้อนเมฆและ สภาพอากาศ

$$P_{mea}(t) = P_{clearsky}(t) + P_{stochastic}(t) \quad (2.31)$$

โดยที่ $P_{clearsky}(t)$ คือกำลังผลิตหรือความเข้มแสงที่คาดว่าจะผลิตได้เมื่อท้องฟ้าสดใส

$P_{stochastic}(t)$ คือตัวแปรที่แสดงถึงความแปรปรวนของกำลังผลิต

การคำนวณ Skill Score โดยใช้ Persistence model เป็นแบบจำลองอ้างอิง

Skill score improvement (SS) คือการเปรียบเทียบแบบจำลอง 2 แบบจำลอง โดยที่ค่า ผลลัพธ์จะบ่งบอกว่าแบบจำลองที่เรา กำลังพัฒนาอยู่สามารถให้ผลลัพธ์ดีกว่าแบบจำลองที่นำมา เปรียบเทียบมากน้อยเพียงใดโดยมีการนิยามดังสมการที่ (2.32)

$$Skill\ score = 1 - \frac{RMSE_{Forecast}}{RMSE_{Reference}} \quad (2.32)$$

โดยที่ $RMSE_{Forecast}$ หมายถึงค่า Root Mean Square Error ที่คำนวณจาก แบบจำลองที่เรา กำลังทดสอบ

$RMSE_{Reference}$ หมายถึงค่า Root Mean Square Error ที่คำนวณจาก
แบบจำลองที่ใช้เพื่อเป็นมาตรฐานในการเปรียบเทียบกับ
แบบจำลองที่เรากำลังทดสอบ

โดยที่ผลลัพธ์ค่า skill score จะมีค่าระหว่าง 1 กับ 0 ถ้าผลลัพธ์มีค่าใกล้ 1 แสดงว่า
แบบจำลองที่ใช้ทำนายให้ผลลัพธ์ดีกว่าแบบจำลองที่นำมาเปรียบเทียบและถ้าผลลัพธ์เข้าใกล้ 0 แสดง
ว่าแบบจำลองที่ใช้ทำนายแสดงความสามารถในการพัฒนาที่น้อยเมื่อเทียบกับแบบจำลองที่นำมา
เปรียบเทียบ แต่ถ้าค่า skill score ตีลบแสดงว่าแบบจำลองที่ใช้ทำนายมีความสามารถด้อยกว่า
แบบจำลองที่นำมาเปรียบเทียบ

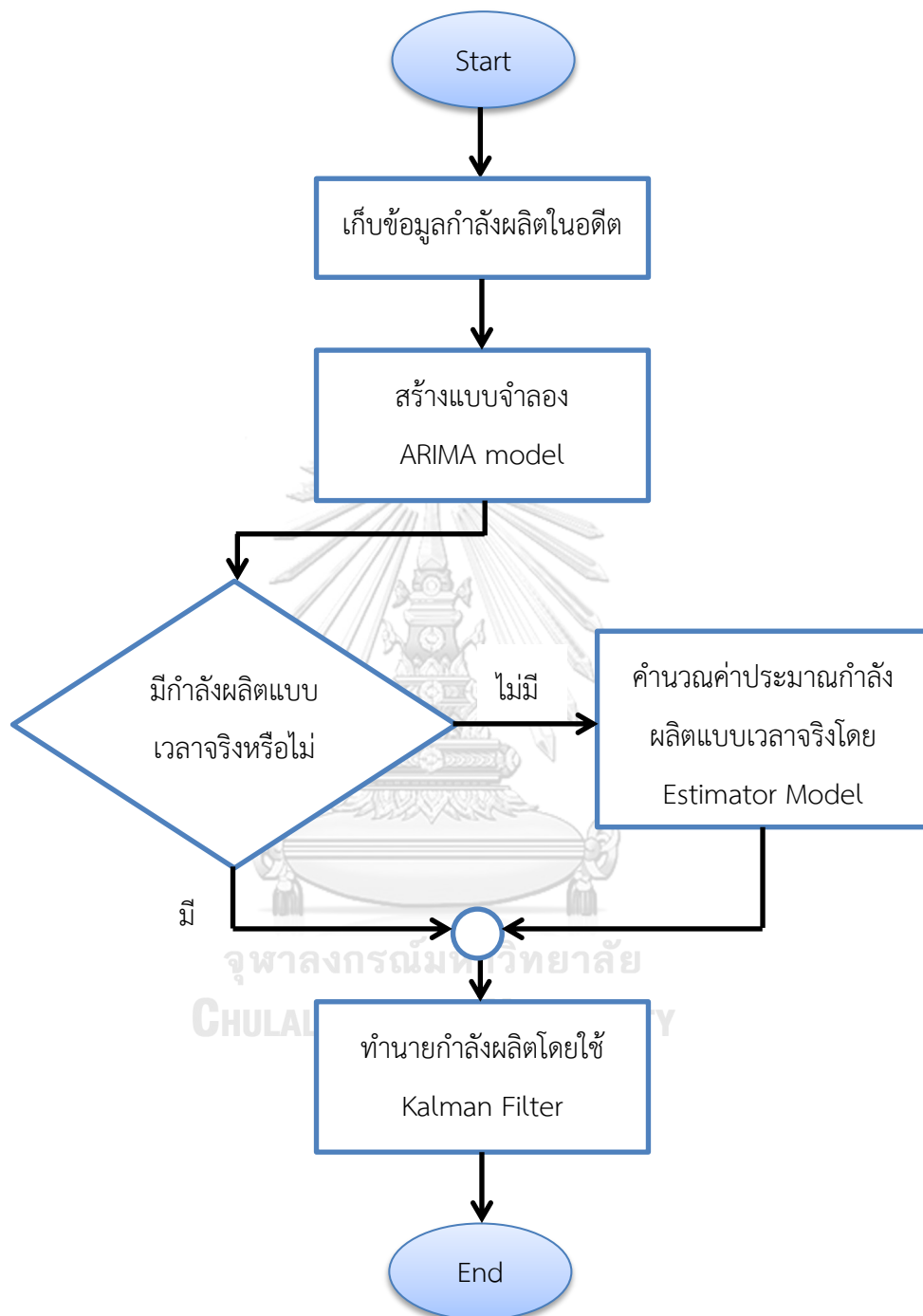


บทที่ 3 แนวทางที่เสนอ

3.1 ภาพรวมขั้นตอนการทำงาน

ในบทนี้จะกล่าวถึงแนวทางที่เสนอในการทำนายและติดตามกำลังผลิตแบบเวลาจริง โดยที่เรามีเป้าหมายที่จะติดตามกำลังผลิตของ PV ที่เราสนใจโดยใช้ข้อมูลกำลังผลิตในอดีตของ PV ที่สนใจ และ PV ข้างเคียงรวมถึงระยะทางระหว่าง PV ที่เราสนใจกับ PV ข้างเคียงมาใช้ในการคำนวณ เพื่อแก้ปัญหาที่เราไม่สามารถทราบข้อมูลกำลังผลิตแบบเวลาจริงของ PV ที่เราสนใจได้อาจเนื่องมาจากความผิดพลาดบางอย่างของระบบผลิตไฟฟ้า แนวทางที่เสนอมิแนวคิดที่จะใช้กำลังผลิตแบบเวลาจริงของ PV ข้างเคียงมาใช้ในการติดตามกำลังผลิตของ PV ที่สนใจโดยผ่าน Kalman Filter อัลกอริทึม ซึ่งอัลกอริทึมนี้ประกอบไปด้วยขั้นตอน 2 ขั้นตอนใหญ่ๆคือ prediction และ correction โดยในขั้นตอน prediction อัลกอริทึมจะใช้ state space model ในการคำนวณค่าถัดไปของกำลังผลิต จากนั้นอัลกอริทึมจะใช้ค่าวัดกำลังผลิตแบบเวลาจริงในการปรับแก้ค่าจากการคำนวณ ดังนั้นแนวทางที่เสนอจะเริ่มจากการเก็บข้อมูลกำลังผลิตในอดีตของ PV ที่เราสนใจมาวิเคราะห์โดยใช้ ARIMA model เพื่อใช้สร้าง state space model หลังจากนั้น จะใช้ข้อมูลกำลังผลิตของ PV ข้างเคียงมาคำนวณค่าประมาณกำลังผลิตที่เราสมมติว่าได้จากการวัดจริงในสมการของ Estimator model โดยการคำนวณทั้ง 2 ส่วนนี้จะใช้ในอัลกอริทึมของ Kalman filter เพื่อคำนวณกำลังผลิตของ PV ที่สนใจแบบเวลาจริง โดยสามารถเขียนกระบวนการเป็น Flowchart ได้ดังรูปที่ 3-1

รูปที่ 3-1 แสดงภาพรวมขั้นตอนการทำงานของวิธีการที่นำเสนอ โดยเริ่มต้นด้วยการเก็บข้อมูลในอดีตของสถานี PV ที่สนใจรวมถึงเก็บข้อมูลกำลังผลิตของสถานี PV รอบข้างและระยะทางระหว่าง PV ที่สนใจไปยัง PV รอบข้าง จากนั้นนำข้อมูลกำลังผลิตในอดีตไปสร้างแบบจำลอง ARIMA model เพื่อเก็บค่า coefficient ของ ARIMA model ซึ่งค่า coefficient นี้จำเป็นต่อการคำนวณในกระบวนการ Kalman Filter จากนั้นเราตรวจสอบว่าสถานี PV ที่เราสนใจมีกำลังผลิตแบบเวลาจริงหรือไม่ หรือก็คือสถานี PV ที่เราสนใจยังมีการติดต่อหรือส่งข้อมูลกำลังผลิตได้หรือไม่ ในกรณีที่เราไม่สามารถทราบกำลังผลิตแบบเวลาจริงได้เราจะใช้ Estimator model ประมาณกำลังผลิตแบบเวลาจริงเพื่อใช้ในการคำนวณของ Kalman Filter ในขั้นตอนสุดท้ายเราจะทำนายหรือติดตามกำลังผลิตของ PV ที่เราสนใจโดยใช้กระบวนการ Kalman Filter



รูปที่ 3-1 ภาพรวมการทำงานของระบบ

3.2 การเก็บข้อมูล

ข้อมูลของโซลาร์เซลล์เราสามารถรวบรวมได้จาก www.pvoutput.org เมื่อเข้าไปยังหน้า homepage ของเว็บไซต์แล้วสมัครสมาชิกเพื่อ log-in เข้าไปยังเว็บไซต์ดังรูปที่ 3-2 เมื่อ log-in แล้วให้เลือกหัวข้อด้านบนที่ชื่อว่า Statistics แล้วให้เราเลือกประเทศที่ต้องการข้อมูลของโซลาร์เซลล์ดังรูปที่ 3-3 และ 3-4 ตามลำดับ จากนั้นเราจะเลือกสถานีโซลาร์เซลล์ที่เราจะนำมาวิเคราะห์ดังรูปที่ 3-5

รูปที่ 3-2 homepage ของ www.pvoutput.org

CHULALONGKORN UNIVERSITY

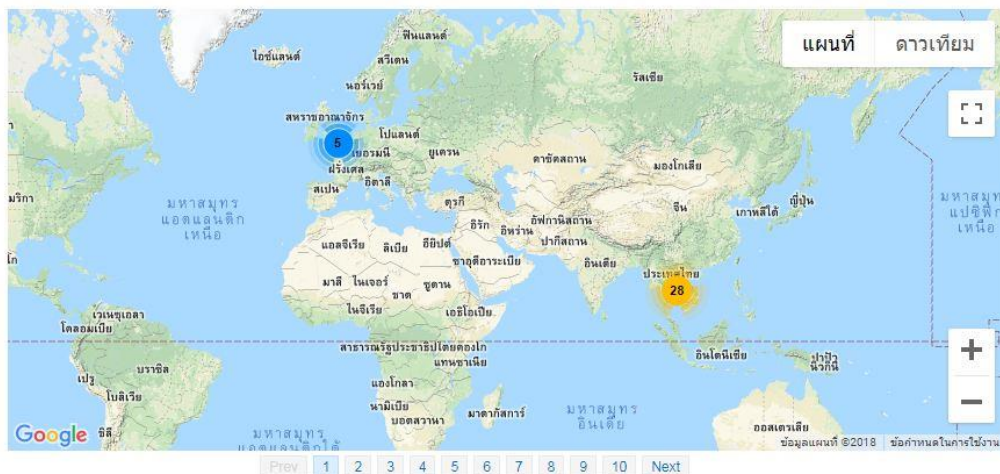
รูปที่ 3-3 แถบเครื่องมือของเว็บไซต์

Country Statistics

Rank	Country	Generated	Share	Average	Capacity	Share	Systems	Average Size	Efficiency
1	Australia	184.640GWh	43.24%	20.250kWh	73.117MW	39.13%	12,387	5.902kW	3.431kWh/kW
2	Netherlands	65.689GWh	15.38%	11.568kWh	34.776MW	18.61%	7,670	4.534kW	2.551kWh/kW
3	United States	54.234GWh	12.70%	27.759kWh	18.504MW	9.90%	2,368	7.814kW	3.552kWh/kW
4	Italy	33.296GWh	7.80%	27.942kWh	11.753MW	6.29%	1,148	10.237kW	2.730kWh/kW
5	United Kingdom	21.183GWh	4.96%	9.657kWh	10.171MW	5.44%	2,455	4.142kW	2.331kWh/kW
6	Germany	20.720GWh	4.85%	17.985kWh	10.688MW	5.72%	1,411	7.574kW	2.375kWh/kW
7	Belgium	15.778GWh	3.69%	16.348kWh	7.426MW	3.97%	1,158	6.412kW	2.550kWh/kW
8	Denmark	5.901GWh	1.38%	11.272kWh	2.520MW	1.35%	555	4.540kW	2.483kWh/kW
9	Thailand	3.586GWh	0.84%	38.189kWh	6.537MW	3.50%	191	34.224kW	1.116kWh/kW
10	Greece	3.450GWh	0.81%	64.663kWh	1.514MW	0.81%	85	17.814kW	3.630kWh/kW
11	Spain	2.797GWh	0.65%	18.685kWh	1.281MW	0.69%	171	7.491kW	2.494kWh/kW
12	Sweden	2.250GWh	0.53%	24.330kWh	1.402MW	0.75%	137	10.236kW	2.377kWh/kW
13	Israel	2.043GWh	0.48%	73.204kWh	229.605kW	0.12%	15	15.307kW	4.782kWh/kW
14	China	1.948GWh	0.46%	201.272kWh	875.145kW	0.47%	26	33.659kW	5.980kWh/kW
15	Switzerland	1.837GWh	0.43%	19.022kWh	907.545kW	0.49%	116	7.823kW	2.432kWh/kW
16	Canada	1.829GWh	0.43%	22.744kWh	667.160kW	0.36%	95	7.022kW	3.239kWh/kW
17	Malta	1.590GWh	0.37%	18.713kWh	1.016MW	0.54%	232	4.377kW	4.275kWh/kW
18	Hungary	1.562GWh	0.37%	48.087kWh	1.626MW	0.87%	123	13.221kW	3.637kWh/kW
19	Cyprus	1.439GWh	0.34%	122.989kWh	1.298MW	0.69%	49	26.486kW	4.644kWh/kW
20	Austria	1.251GWh	0.29%	11.367kWh	528.908kW	0.28%	117	4.520kW	2.515kWh/kW

รูปที่ 3-4 หน้าต่างเลือกประเทศที่ต้องการเพื่อดูข้อมูลของสถานีโซลาร์เซลล์

Add Output | Your Outouts | PV Ladder | Statistics | Live Outouts | Teams | Favourites | Settings | Community | Donations | Help | Logout

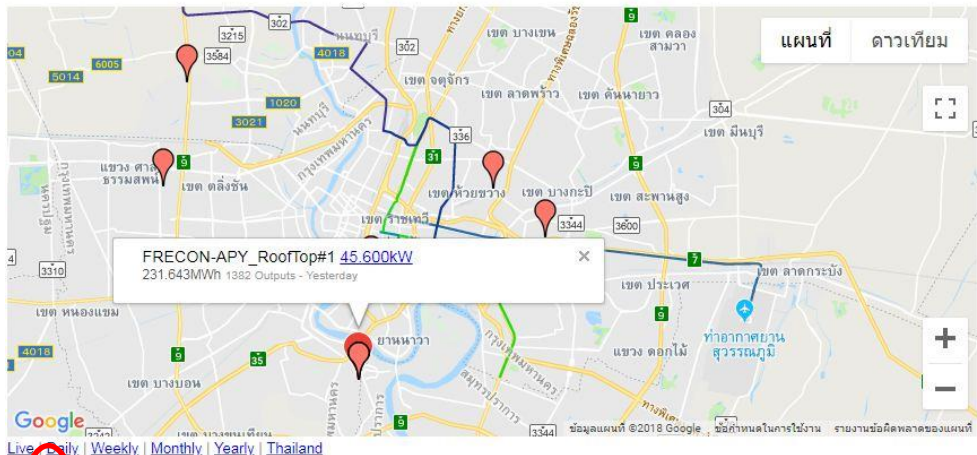


Thailand 6.537MW

Find: Tips

Rank	Name	Location	System Size	Generation	Efficiency	Average	Outputs
1	FRECON-Multifiber	Thailand	412.920kW	921.217MWh	3.056kWh/kW	1,261.941kWh	730 Days
2	Honda R&D Southeast Asia	Thailand	108.000kW	479.964MWh	3.357kWh/kW	362.510kWh	1,324 Days
3	FRECON-HeatLava	Thailand	153.140kW	288.380MWh	2.649kWh/kW	405.598kWh	711 Days
4	FRECON-APY_Rooftop#1	Thailand	45.600kW	231.643MWh	3.676kWh/kW	167.614kWh	1,382 Days
5	FRECON-APY_Rooftop#2	Thailand	82.460kW	197.826MWh	2.476kWh/kW	204.155kWh	969 Days
6	Science Building	Thailand	25.575kW	110.377MWh	3.985kWh/kW	101.918kWh	1,083 Days
7	us solar1	Thailand	63.000kW	47.599MWh	3.935kWh/kW	247.913kWh	192 Days
8	Honda R&D Southeast Asia 2	Thailand	10.000kW	44.261MWh	3.458kWh/kW	34.579kWh	1,280 Days
9	Solar D Home - Archawin	Thailand	9.900kW	43.129MWh	3.827kWh/kW	38.037kWh	1,134 Days

รูปที่ 3-5 หน้าต่างรายชื่อสถานีโซลาร์เซลล์ที่มีข้อมูลในประเทศที่เลือก



FRECON-APY_Rooftop#1 45.600kW Find: [Tips](#)

Distance	Name	Location	System Size	Generation	Efficiency	Average	Outputs	Actions
0.00km	FRECON - APY2	Thailand	11.200kW	8.587MWh	4.406kWh/kW	49.348kWh	174 Days	Compare Add
0.00km	FRECON - APY4	Thailand	11.800kW	8.221MWh	4.409kWh/kW	52.030kWh	158 Days	Compare Add
0.01km	FRECON - APY3	Thailand	11.500kW	7.893MWh	4.344kWh/kW	49.954kWh	158 Days	Compare Add
0.01km	FRECON - APY1	Thailand	44.800kW	24.222MWh	1.148kWh/kW	51.428kWh	471 Days	Compare Add
7.02km	Solar D Home - Vittaya1	Thailand	4.800kW	3.692MWh	4.446kWh/kW	21.341kWh	173 Days	Compare Add
9.27km	Bangkok Soladin 600	Thailand	0.650kW	3.218MWh	3.405kWh/kW	2.213kWh	1,454 Days	Compare Add
15.30km	Bsq 600-69	Thailand	6.000kW	22.150MWh	3.786kWh/kW	22.718kWh	975 Days	Compare Add
15.34km	FRECON-Pornicha1	Thailand	11.780kW	38.449MWh	3.389kWh/kW	39.926kWh	963 Days	Compare Add
18.03km	Solar D Home - Prayut	Thailand	3.780kW	6.694MWh	3.979kWh/kW	15.042kWh	445 Days	Compare Add
21.53km	Solar D - Prempracha	Thailand	24.200kW	36.833MWh	3.548kWh/kW	85.859kWh	429 Days	Compare Add

รูปที่ 3-6 ตารางแสดงรายชื่อและระยะทางของสถานีรอบข้างกับสถานีที่เลือก

เมื่อเราเลือกสถานีที่ต้องการได้แล้วให้เรากดเข้าไปดูรายละเอียดข้างในสถานีโดยกดที่ชื่อของสถานีนั้นๆ เมื่อกดแล้วจะได้ตามรูปที่ 3-6 แผนที่ในรูปที่ 3-6 จะบอกสถานที่ตั้งของสถานีโซลาร์เซลล์ เราจะต้องเก็บค่าระยะทางระหว่างสถานีต่างๆที่มีรายละเอียดอยู่ที่ในตารางคอลัมน์ Distance เพื่อใช้ในการคำนวณเพื่อประมาณค่ากำลังผลิตของโซลาร์เซลล์ใน Estimator model จากนั้นให้เรากดเข้าไปที่ปุ่ม Live ที่มุมซ้ายล่างข้างใต้แผนที่เพื่อเข้าไปเก็บข้อมูลรายละเอียดกำลังผลิตของโซลาร์เซลล์ โดยสามารถดูข้อมูลย้อนหลังสูงสุดได้ 2 อาทิตย์ และสามารถดูข้อมูลแบบเวลาจริงได้

จากรูปที่ 3-7 เราจะเก็บค่าในคอลัมน์ Normalized สาเหตุที่เก็บค่า Normalized เนื่องจากเราจะวิเคราะห์สถานีโซลาร์เซลล์ที่มีกำลังผลิตต่างกัน ดังนั้นเราจึงใช้การ Normalized ค่าของกำลังผลิตของแต่ละสถานีในการวิเคราะห์เพื่อทำให้สามารถนำข้อมูลกำลังผลิตมาคำนวณร่วมกันหรือเปรียบเทียบกันได้ โดยที่สมการ Normalized เป็นไปดังสมการที่ (3.1)

$$P_{Normalized,t} = \frac{P_{Average,t}}{P_{Max}} \tag{3.1}$$

โดยที่ $P_{Normalized,t}$ คือกำลังผลิตที่ถูก normalized ที่เวลา t สามารถทราบได้จากคอลัมน์ Normalized ในรูปที่ 3-7


$P_{Average,t}$ คือกำลังผลิตเฉลี่ยที่เวลา t สามารถทราบได้จากคอลัมน์ Average ในรูปที่ 3-7

P_{Max} คือกำลังผลิตสูงสุดของแต่ละสถานีโซลาร์เซลล์ สามารถทราบได้จากค่า กำลังผลิตหลังชื่อของสถานีในรูปที่ 3-7

เมื่อเก็บข้อมูลเสร็จแล้วเราจะได้ ตารางแสดงกำลังผลิตของสถานีโซลาร์เซลล์แต่ละสถานีที่ Normalized ค่าแล้วสอดคล้องกับเวลาในแต่ละวันและชุดข้อมูลระยะทางระหว่างสถานีโซลาร์เซลล์แต่ละแห่งจากนั้นจึงเริ่มวิเคราะห์ข้อมูล

Live | Daily | Weekly | Monthly | Yearly | Analyse | Map | Message | Favourite | Insolation | Customise | Minimise | Refresh Off

Generation **97 of 31746** · 4 Followers · 1 Following · 0 Photos · **232 MWh** · **247.9T** CO₂ · 2 Systems

Target  121% · **\$109.29** ▲ · **182.15** Wh · **32,000W** · **40,520W** Peak · 0 to 71C

FRECON-APY_RoofTop#1 45.600kW Compare: Tips

Date	Time	Energy	Efficiency	Power	Average	Normalised	Temperature	Voltage	Energy Used	Power Used
27/09/18	2:05PM	182.150kWh	3.995kWh/kW	32.000W	32.004W	0.702kW/kW	59.0C	398.0V	-	-
27/09/18	2:00PM	179.483kWh	3.936kWh/kW	31.360W	31.356W	0.688kW/kW	56.0C	399.0V	-	-
27/09/18	1:55PM	176.870kWh	3.879kWh/kW	30.810W	30.816W	0.676kW/kW	53.0C	397.5V	-	-
27/09/18	1:50PM	174.302kWh	3.822kWh/kW	31.560W	31.560W	0.692kW/kW	55.0C	394.2V	-	-
27/09/18	1:45PM	171.672kWh	3.765kWh/kW	31.380W	31.380W	0.688kW/kW	58.0C	392.8V	-	-
27/09/18	1:40PM	169.057kWh	3.707kWh/kW	31.590W	31.596W	0.693kW/kW	60.0C	396.0V	-	-
27/09/18	1:35PM	166.424kWh	3.650kWh/kW	32.510W	32.508W	0.713kW/kW	60.0C	396.8V	-	-
27/09/18	1:30PM	163.715kWh	3.590kWh/kW	33.550W	33.552W	0.736kW/kW	61.0C	397.8V	-	-
27/09/18	1:25PM	160.919kWh	3.529kWh/kW	34.500W	34.500W	0.757kW/kW	59.0C	396.2V	-	-
27/09/18	1:20PM	158.044kWh	3.466kWh/kW	39.680W	39.684W	0.870kW/kW	53.0C	397.0V	-	-
27/09/18	1:15PM	154.737kWh	3.393kWh/kW	12.260W	12.264W	0.269kW/kW	47.0C	392.2V	-	-
27/09/18	1:10PM	153.715kWh	3.371kWh/kW	8.530W	8.532W	0.187kW/kW	50.0C	390.8V	-	-
27/09/18	1:05PM	153.004kWh	3.355kWh/kW	14.800W	14.796W	0.324kW/kW	59.0C	395.8V	-	-
27/09/18	1:00PM	151.771kWh	3.328kWh/kW	13.060W	13.056W	0.286kW/kW	64.0C	396.2V	-	-
27/09/18	12:55PM	150.683kWh	3.304kWh/kW	34.000W	33.996W	0.746kW/kW	71.0C	401.8V	-	-
27/09/18	12:50PM	147.850kWh	3.242kWh/kW	33.980W	33.984W	0.745kW/kW	70.0C	401.2V	-	-
27/09/18	12:45PM	145.018kWh	3.180kWh/kW	34.820W	34.824W	0.764kW/kW	67.0C	399.0V	-	-
27/09/18	12:40PM	142.116kWh	3.117kWh/kW	34.590W	34.596W	0.759kW/kW	66.0C	397.2V	-	-
27/09/18	12:35PM	139.233kWh	3.053kWh/kW	34.740W	34.740W	0.762kW/kW	70.0C	394.0V	-	-
27/09/18	12:30PM	136.338kWh	2.990kWh/kW	37.050W	37.056W	0.813kW/kW	65.0C	399.2V	-	-
27/09/18	12:25PM	133.250kWh	2.922kWh/kW	37.580W	37.584W	0.824kW/kW	64.0C	398.0V	-	-
27/09/18	12:20PM	130.118kWh	2.853kWh/kW	33.930W	33.936W	0.714kW/kW	62.0C	397.0V	-	-
27/09/18	12:15PM	127.290kWh	2.791kWh/kW	36.530W	36.528W	0.811kW/kW	66.0C	400.0V	-	-
27/09/18	12:10PM	124.246kWh	2.725kWh/kW	36.680W	36.684W	0.804kW/kW	65.0C	400.2V	-	-
27/09/18	12:05PM	121.189kWh	2.658kWh/kW	37.040W	37.044W	0.811kW/kW	65.0C	401.8V	-	-
27/09/18	12:00PM	118.102kWh	2.590kWh/kW	37.810W	37.814W	0.829kW/kW	63.0C	402.0V	-	-
27/09/18	11:55AM	114.951kWh	2.521kWh/kW	22.680W	22.684W	0.497kW/kW	56.0C	393.0V	-	-

รูปที่ 3-7 ตารางแสดงข้อมูลกำลังผลิตและพลังงานที่โซลาร์เซลล์ผลิตได้ของสถานีที่เลือกที่เวลาต่างๆใน 1 วันโดยข้อมูลจะอัปเดตทุก 5 นาที

3.3 การออกแบบ Estimator Model

ในขั้นต่อไปเราจะออกแบบโมเดลในการใช้ค่าที่เก็บมาจากโซลาร์เซลล์ที่อยู่รอบ บริเวณพื้นที่ที่เราต้องการจะทำนายกำลังผลิตจากนั้นจะใช้สมการทางคณิตศาสตร์อธิบายความสัมพันธ์ของค่ากำลังผลิตที่เก็บมาจากสถานีโซลาร์เซลล์ข้างเคียงตามสมการที่ (3.2) (เพื่ออธิบายให้เห็นภาพมากขึ้น เราจะใช้สถานีโซลาร์เซลล์จริงจำนวน 3 สถานีที่ตั้งอยู่ในประเทศไทย)

เป้าหมายของโมเดลนี้คือการใช้กำลังผลิตของ PV ที่ตั้งอยู่รอบ PV ที่เราต้องการทำนายกำลังผลิตเพื่อประมาณกำลังผลิตแบบเวลาจริงและนำไปใช้ในกระบวนการ Kalman filter แนวคิดของโมเดลนี้เกิดมาจากการที่เราวาดกราฟดูชุดข้อมูลกำลังผลิตของทั้ง 3 สถานี (ยกตัวอย่างในรูปที่ 3-8 เส้นกราฟ real-Bsq, real-Apy และ real-Pornticha) พบว่าในบางช่วงเวลาเดียวกันกราฟแต่ละสถานีมีลักษณะแนวโน้มเป็นไปในทิศทางเดียวกันและถ้าสภาพอากาศของแต่ละสถานีเหมือนกันกราฟกำลังผลิตน่าจะมีรูปร่างคล้ายคลึงกันด้วย จึงตั้งสมมติฐานว่ากำลังผลิตของทั้ง 3 สถานีน่าจะมีความเกี่ยวข้องกันโดยมีตัวแปรระยะทางเข้ามาเกี่ยวข้อง

$$est_{i,t} = \left(\frac{d_{ij}}{d_{ij} + d_{ik} + \dots + d_{in}} \right) (mea_{j,t}) + \left(\frac{d_{ik}}{d_{ij} + d_{ik} + \dots + d_{in}} \right) (mea_{k,t}) + \dots + \left(\frac{d_{in}}{d_{ij} + d_{ik} + \dots + d_{in}} \right) (mea_{n,t}) \quad (3.2)$$

โดยที่

$est_{i,t}$ คือ ค่าประมาณของกำลังผลิตของโซลาร์เซลล์ที่สถานี i ที่เราสนใจที่เวลา t

$mea_{j,t}$ คือ ค่ากำลังผลิตของสถานี j ที่ได้จากการวัดจริงที่เวลา t

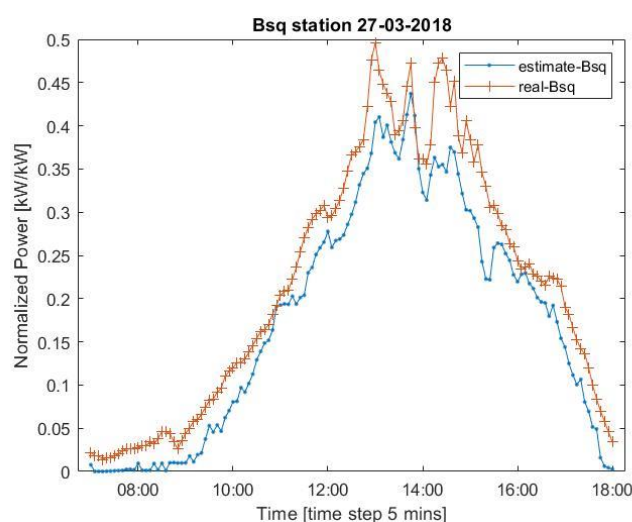
$mea_{k,t}$ คือ ค่ากำลังผลิตของสถานี k ที่ได้จากการวัดจริงที่เวลา t

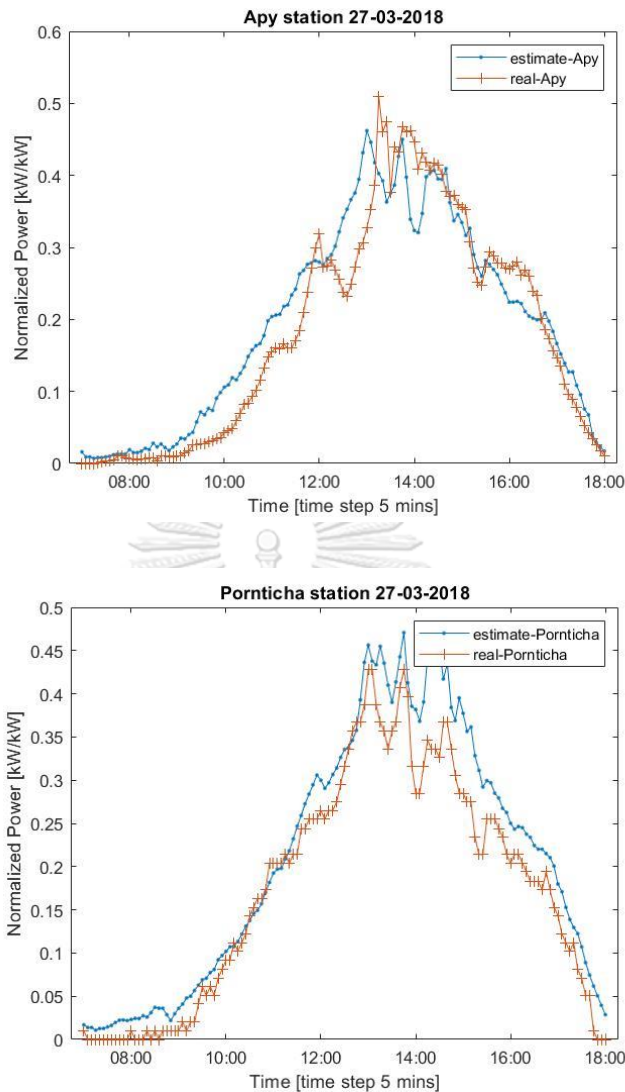
$mea_{n,t}$ คือ ค่ากำลังผลิตของสถานี n ที่ได้จากการวัดจริงที่เวลา t

d_{ij} คือ ส่วนกลับของระยะทางระหว่างสถานี i กับ j

จากสมการที่ (3.2) เราจะประมาณค่ากำลังผลิตแบบเวลาจริงของสถานี PV ที่เราสนใจโดยการนำกำลังผลิตของแต่ละสถานีคูณด้วยอัตราส่วนของระยะทาง ยกตัวอย่างเช่น PV i คือ PV ที่เราสนใจ เทอมแรกในสมการที่ (3.2) จะคำนวณอัตราส่วนของกำลังผลิตที่เกิดจาก PV j โดยการคูณ “กำลังผลิตของ PV j ” ซึ่งก็คือตัวแปร $mea_{j,t}$ ด้วยอัตราส่วนระหว่าง “ส่วนกลับของระยะทางระหว่าง PV i กับ j ” ซึ่งก็คือตัวแปร d_{ij} ทหารด้วย “ส่วนกลับของระยะทางระหว่างแต่ละ PV ไปยัง PV i รวมกัน” ฉะนั้นจากสมการจะสังเกตได้ว่า PV ที่อยู่ใกล้ PV i มากกว่าจะส่งผลกับผลลัพธ์กำลังผลิตที่ถูกประมาณค่าของ PV i มากกว่า PV ที่อยู่ห่างไกลออกไป

จากนั้นเพื่อลดจำนวนเทอมในสมการและเวลาการคำนวณ เราจะวิเคราะห์กราฟกำลังผลิตโดยใช้หลักการของค่า correlation coefficient ในการพิจารณาว่าสถานีโซลาร์เซลล์แต่ละแห่งมีความสัมพันธ์ต่อกันอย่างไรบ้างโดยการนำข้อมูลกำลังผลิตของสถานีโซลาร์เซลล์ทั้ง 3 แห่งมาวาดกราฟพิจารณาลักษณะของกราฟในวันเดียวกันและนำข้อมูลกำลังผลิตในแต่ละวันมาคำนวณหาค่า correlation coefficient เพื่อดูความสัมพันธ์ของค่า correlation coefficient (R) และ p-value (P) ในแต่ละวัน ซึ่งค่า P คือผลลัพธ์ที่ได้จากการคำนวณและเป็นค่าที่ระบุว่าคุณค่า R ที่ตำแหน่งเดียวกันนั้นมีความน่าเชื่อถือหรือไม่ เมื่อเราได้ค่า R และ P แล้วจะทำให้เราสามารถตัดสินใจได้ว่าจะใช้ค่าที่เก็บได้จากสถานีโซลาร์เซลล์ใดบ้างในการคำนวณ





รูปที่ 3-8 กราฟเปรียบเทียบค่าประมาณที่ได้จากโมเดลประมาณค่าและค่าที่ได้จากการวัดจริงของสถานีทั้ง 3 สถานี

จากรูปกราฟทั้ง 3 ภาพในรูปที่ 3-8 ได้ยกตัวอย่างกำลังผลิตของวันที่ 27 มีนาคม พ.ศ. 2561 เพื่อให้เห็นภาพมากขึ้น กราฟทั้ง 3 ได้แสดงผลลัพธ์จากการคำนวณโดยใช้โมเดลประมาณค่าตามสมการที่ (3.2) ของ สถานี Apy, Bsq และ Pornticha เปรียบเทียบกับกำลังผลิตของจริงที่เวลาจริง จะเห็นได้ว่ารูปร่างของกราฟมีลักษณะที่คล้ายคลึงกับค่ากำลังผลิตของจริง แสดงให้เห็นว่าค่าที่คำนวณได้จากสมการ Estimator model มีความใกล้เคียงกับกำลังผลิตของจริง

ประกอบกับเมื่อเราวิเคราะห์ค่า correlation coefficient เพื่อยืนยันความสัมพันธ์ของสถานีทั้ง 3 สถานีที่ยกตัวอย่าง (ค่าที่ได้เป็นค่าที่ได้จากการคำนวณในวันที่ 27 มีนาคม พ.ศ. 2561)

(R27 คือ ค่า correlation coefficient ระหว่าง 3 สถานีในวันที่ 27 มีนาคม พ.ศ.2561 และ P27 คือ ค่า p-value test ระหว่าง 3 สถานีในวันเดียวกัน)

R27			P27		
ans =			ans =		
1.0000	0.9520	0.9846	1.0000	0.0000	0.0000
0.9520	1.0000	0.9417	0.0000	1.0000	0.0000
0.9846	0.9417	1.0000	0.0000	0.0000	1.0000

จากเมทริกซ์ที่คำนวณได้ แถวและคอลัมน์แรกแทนสถานี Bsq แถวและคอลัมน์ที่สองแทนสถานี Apy แถวและคอลัมน์สุดท้ายแทนสถานี Pornticha ค่าที่อยู่ในเมทริกซ์ R27 เป็นค่า Pearson correlation coefficient ซึ่งจากผลลัพธ์ในตารางมีค่าประมาณ 0.9 (ไม่นับค่าในแนว diagonal) ซึ่งสามารถสรุปได้ว่าค่ากำลังผลิตของแต่ละสถานีในวันที่ 27 มีนาคม พ.ศ. 2561 มีความสัมพันธ์แบบเป็นไปในทิศทางเดียวกันและค่าที่อยู่ในเมทริกซ์ P27 คือค่า p-value test ในกรณีของการหาค่า correlation จะมี null hypothesis คือชุดข้อมูลของตัวแปรซึ่งถูกนำมาคำนวณหาค่า correlation ทั้ง 2 ชุดข้อมูลไม่มีความสัมพันธ์ต่อกัน ซึ่งถ้าหากค่า p-value test มีค่าน้อยกว่า 0.05 (ไม่นับค่าในแนว diagonal) แสดงว่าเราปฏิเสธ null hypothesis หรือก็คือชุดข้อมูลทั้ง 2 ชุดมีความสัมพันธ์กันอย่างมีนัยสำคัญ

สรุปได้ว่าสมมติฐานที่ว่าค่ากำลังผลิตของทั้ง 3 สถานีมีความเกี่ยวข้องกันนั้นถูกต้องอย่างมีนัยสำคัญ ดังนั้นจึงสามารถสรุปได้ว่าโมเดลประมาณค่าตามสมการที่ (3.2) สามารถนำมาประมาณค่ากำลังผลิตได้ เนื่องจากกำลังผลิตของทั้ง 3 สถานีมีความเกี่ยวข้องกันจากการที่เราวิเคราะห์ความสัมพันธ์ผ่านค่า correlation coefficient

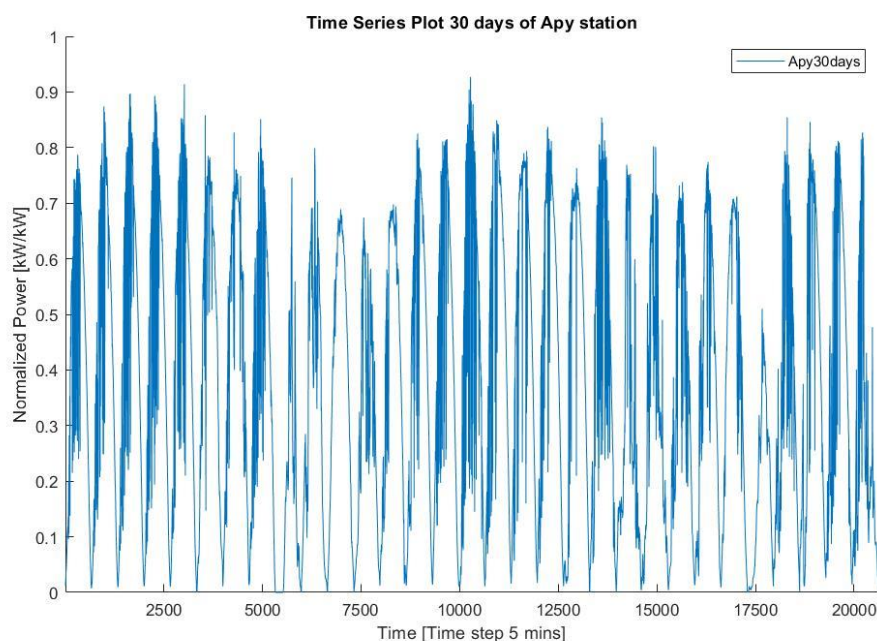
3.4 การใช้ ARIMA Model เพื่อสร้าง State Space Model

ในส่วนของแบบจำลอง ARIMA เราจะใช้ ARIMA เพื่อจำลองกราฟให้ได้ใกล้เคียงกับชุดข้อมูลที่เราเก็บมาโดยการทำให้ differencing ข้อมูลเพื่อให้ชุดข้อมูลมีความเป็น stationary โดยสามารถสังเกตได้จากกราฟ ACF หลังจากที่ทำ differencing แล้วกราฟจะมีลักษณะที่ลดลงอย่างรวดเร็วกว่าก่อนที่จะทำ differencing จากนั้นจะใช้ทฤษฎีของ Box & Jenkins ในการหาค่าพารามิเตอร์ p, d และ q ของแบบจำลอง ARIMA เมื่อสร้างแบบจำลองเสร็จแล้วจะพิจารณาจาก residual plot ว่า

แบบจำลองที่ได้เหมาะสมหรือไม่ โดยแบบจำลองที่เหมาะสม จะมี residual histogram เป็นลักษณะ normally distributed และ เมื่อ plot ACF แล้วปรากฏว่าไม่มี correlation กันระหว่างลำดับ lag หรือมีค่า correlation เล็กน้อย ดังที่ได้อธิบายไว้ในบทที่ 2 หัวข้อที่ 2.1 จากนั้นเราจะเขียนสมการแบบจำลอง ARIMA ในรูปแบบของ state-space model เพื่อเตรียมพร้อมสำหรับใช้ในกระบวนการของ Kalman filter

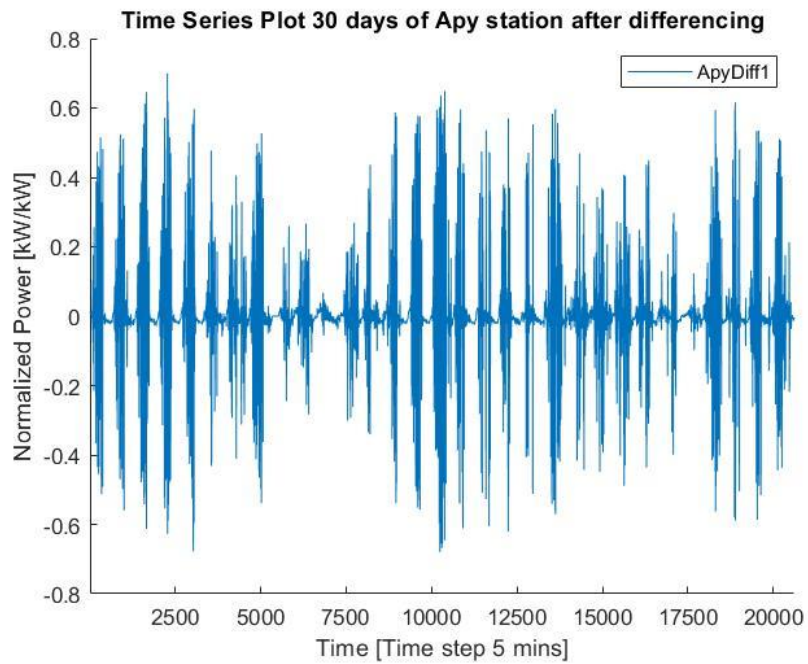
ในขั้นแรกข้อมูลที่เก็บจากเว็บไซต์จะเป็นข้อมูลรายวัน มีระยะห่างระหว่างข้อมูลทุก 5 นาที ข้อมูลกำลังผลิตที่เราเก็บจะเป็นข้อมูลกำลังผลิตที่ผ่านการ normalized เรียบร้อยแล้ว จากนั้นเราจะเลือก PV 1 แห่งในระบบเพื่อเลือกเป็น PV ที่ต้องการทดสอบทำนายค่ากำลังผลิต ส่วน PV ที่เหลือจะใช้ในการคำนวณ Estimator model เราได้ยกตัวอย่างชุดข้อมูลที่เก็บมาแสดงในรูปที่ 3-9 ขั้นตอนต่อไปจะวิเคราะห์ชุดข้อมูลตัวอย่างเพื่อสร้างแบบจำลอง ARIMA โดยใช้ตัวอย่างในรูปที่ 3-9

จากตัวอย่างในรูปที่ 3-9 จะเห็นได้ว่าชุดข้อมูลยังไม่เป็น stationary ดังนั้นเราจะทำ differencing ข้อมูล 1 ครั้ง ดังรูปที่ 3-10 จากนั้นวิเคราะห์กราฟโดยการใช ACF และ PACF จะได้ผลลัพธ์ตามรูปที่ 3-11 และ 3-12

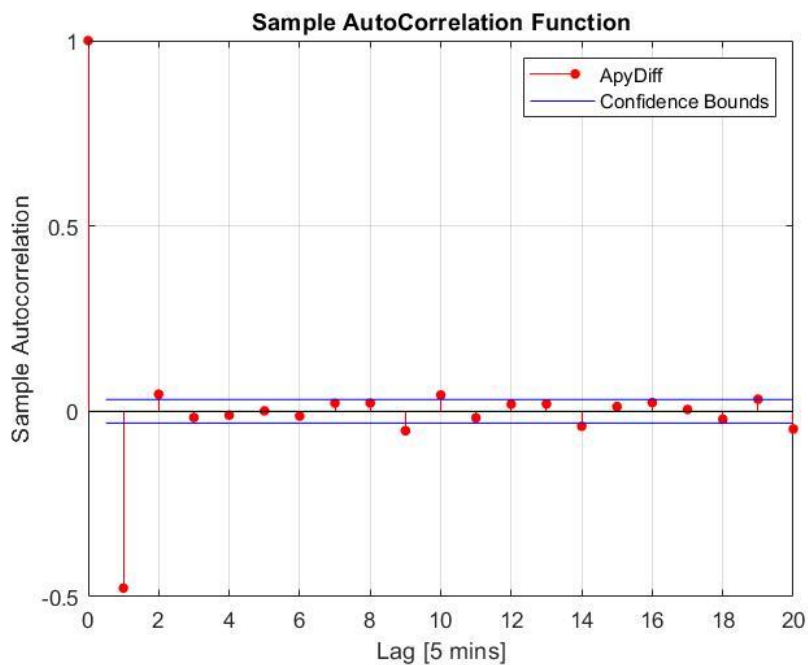


รูปที่ 3-9 รูปชุดข้อมูลตัวอย่างของกำลังผลิตทั้งหมด 30 วันของสถานี Apy ในเดือน

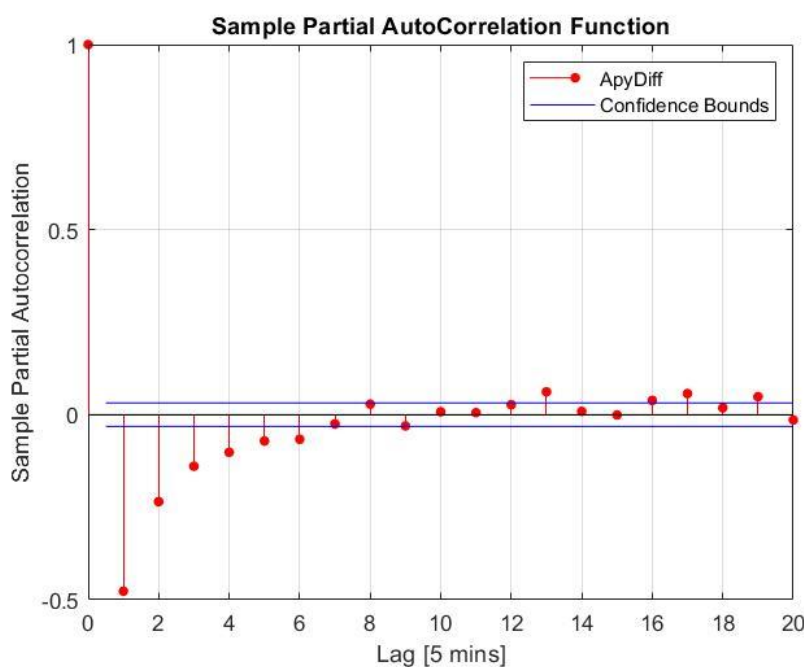
มีนาคม พ.ศ.2561



รูปที่ 3-10 รูปชุดข้อมูลตัวอย่างหลังการทำ differencing 1 ครั้ง



รูปที่ 3-11 รูปแสดงการกราฟ ACF ของชุดข้อมูลตัวอย่าง



รูปที่ 3-12 รูปแสดงการกราฟ PACF ของชุดข้อมูลตัวอย่าง

โดยกราฟ ACF และ PACF จะมีค่าในแกนนอนเป็นจำนวน lag ต่างๆที่ใช้คำนวณค่า autocorrelation coefficient เมื่อค่า lag มีค่านั้นๆ เช่นเมื่อพิจารณาที่ข้อมูลมีระยะเวลาห่างกันเท่ากับ lag 1 แล้วจะคำนวณค่า autocorrelation coefficient ได้เท่ากับ -0.639 ดังรูปที่ 3-11 เมื่อพิจารณากราฟแล้วจะเห็นว่ากราฟ ACF มีค่าที่โดดเด่นอยู่ที่ lag 1 และ กราฟ PACF ค่อยๆลู่เข้าหาศูนย์อย่างช้าๆดังนั้นอาจจะสามารถสรุปได้เบื้องต้นจากรายที่ 2-1 ว่าสามารถใช้แบบจำลอง MA(1) เพื่อจำลองชุดข้อมูลนี้ได้เมื่อทำ differencing แล้ว 1 ครั้ง หรือ ARIMA (0,1,1)

จากนั้นให้เราสร้างแบบจำลอง โดยในขั้นตอนนี้เมื่อสร้างแบบจำลองเสร็จแล้ว MATLAB จะแสดง ชื่อของพารามิเตอร์, ค่าของพารามิเตอร์, ค่า Standard error, ค่า t Statistic และค่า P-Value ดังรูปที่ 3-13 โดยให้เราสังเกตที่ค่า P-Value ค่านี้บ่งบอกถึงการทดสอบสมมติฐานของการประมาณค่าพารามิเตอร์ โดยที่ null hypothesis คือค่าพารามิเตอร์ที่ MATLAB คำนวณมีค่าเท่ากับ 0 หรือก็คือค่าของพารามิเตอร์ที่ไม่มีผลกับแบบจำลอง ซึ่งถ้าค่าของ P-Value น้อยกว่า 0.05 หมายความว่าเราปฏิเสธ null hypothesis แสดงว่าค่าพารามิเตอร์ที่ MATLAB ประมาณค่ามีนัยสำคัญ แต่ถ้า P-Value มีค่ามากกว่า 0.05 บ่งบอกว่าค่าพารามิเตอร์ที่ถูกคำนวณมาไม่มีผลต่อแบบจำลอง สามารถนำค่านี้ออกไปจากแบบจำลองหรือให้มีค่าเท่ากับ 0 ได้

Parameters					Goodness of Fit	
Parameter	Value	Standard Error	t Statistic	P-Value	Measure	Value
Constant	3.6492e-06	7.3470e-04	0.0050	0.9960	AIC	-5.2731e+03
MA{1}	-0.6309	0.0091	-69.2360	0	BIC	-5.2543e+03
Variance	0.0156	2.2847e-04	68.2475	0		

รูปที่ 3-13 รูปแสดงรายละเอียดค่าสัมประสิทธิ์และค่าพารามิเตอร์หลังจากที่สร้างแบบจำลอง ARIMA(0,1,1)

นอกจากค่าพารามิเตอร์ที่ได้กล่าวไปข้างต้นแล้ว MATLAB ยังคำนวณค่า Akaike Information Criterion (AIC) และ Bayesian Information Criterion (BIC) ในรูปที่ 3-13 มาให้เพื่อเป็นตัวเลือกในการพิจารณาแบบจำลอง นอกเหนือจากการตรวจสอบ Residual ของแบบจำลองตามที่ได้กล่าวไปใน บทที่ 2 หัวข้อที่ 2.1.7 Diagnostic Checking ซึ่งค่าทั้งสองนี้สามารถเขียนเป็นสมการได้ดังสมการที่ (3.3) และ (3.4)

$$AIC = -2 * \text{Log}L + 2r \quad (3.3)$$

$$BIC = -2 * \text{Log}L + r * \ln(n) \quad (3.4)$$

โดยที่ r มีค่าเท่ากับจำนวนพารามิเตอร์ในแบบจำลองทั้งหมดรวมกัน ซึ่งก็คือจำนวนของ p , q และ constant ใน ARIMA model

n คือจำนวนของ observation ที่นำมา fit ARIMA model

$\text{Log}L$ คือค่าที่เหมาะสมที่สุดของ Maximum likelihood ที่ถูกคำนวณจากการประมาณค่า parameter ของแบบจำลอง

เราพิจารณาเทอมหลังของทั้ง 2 สมการเรียกว่า penalty factor เนื่องจากมีค่า r ซึ่งเป็นค่าที่บ่งบอกถึงจำนวนพารามิเตอร์ในแบบจำลอง ค่า AIC และ BIC มีประโยชน์ในกรณีที่เรากำลังต้องการเลือกแบบจำลองที่มีจำนวนพารามิเตอร์แตกต่างกัน โดยเราจะเลือกแบบจำลองที่ใช้จำนวนพารามิเตอร์น้อยที่สุดและคำนวณค่า AIC และ BIC แล้วได้ค่าผลลัพธ์ต่ำที่สุดให้เป็นแบบจำลองที่เหมาะสมที่สุด

จากนั้นเราจะนำค่าพารามิเตอร์ที่ได้มาสร้างแบบจำลอง state-space เพื่อนำไปใช้กับ Kalman filter ในขั้นตอนถัดไป จากค่าพารามิเตอร์ในรูปแบบที่ 3-13 สามารถเขียนสมการของ ARIMA (0,1,1) ได้ดังสมการที่ (3.5) และจากสมการที่ (3.5) ย้ายข้าง y_{t-1} มาด้านขวาของสมการได้เป็นสมการที่ (3.6)

$$y_t - y_{t-1} = \mu + e_t - \theta_1 e_{t-1} \quad (3.5)$$

$$y_t = y_{t-1} + \mu + e_t - \theta_1 e_{t-1} \quad (3.6)$$

ในคำสั่งการสร้าง state-space model ของ MATLAB มีการนิยามตัวแปร y_t , y_{t-1} และ e_t ไว้อยู่แล้วแต่เนื่องจากยังเหลือตัวแปร y_{t-1} , μ และ e_{t-1} ทำให้ต้องสร้างสมการมารองรับตัวแปรเหล่านี้เพิ่มดังสมการที่ (3.7) และ (3.8) จากนั้นเปลี่ยนตัวแปรสมการที่ (3.6) ให้เขียนในรูปแบบของสมการที่ (3.7) และ (3.8) ได้ดังสมการที่ (3.9) และเมื่อนำสมการมาเรียงกัน เราสามารถเขียนสมการที่ (3.7), (3.8) และ (3.9) ในรูปของ state-space model ได้ดังสมการที่ (3.10)

$$y_{2,t} = y_{2,t-1} \quad (3.7)$$

$$y_{3,t} = e_{1,t} \quad (3.8)$$

$$y_{1,t} = y_{1,t-1} + \mu y_{2,t-1} + e_t - \theta_1 y_{3,t-1} \quad (3.9)$$

โดยที่ $y_{1,t-1} = y_{t-1}$, $y_{2,t-1} = 1$ และ $y_{3,t-1} = e_{1,t-1}$

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \end{bmatrix} = \begin{bmatrix} 1 & \mu & -\theta_1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ y_{3,t-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} e_{1,t} \quad (3.10)$$

3.5 การทำนายโดยใช้ Kalman Filter

ขั้นตอนการทำนายโดยใช้ Kalman filter จะทำหลังจากเราได้สร้างแบบจำลอง ARIMA model และทราบ coefficient ในรูปของเมทริกซ์แล้วในสมการที่ (3.10) จากนั้นเราจะใช้เมทริกซ์นี้

ในสมการ state forecast ยกตัวอย่างเช่น สมการที่ (3.10) เมทริกซ์ $\begin{bmatrix} 1 & \mu & -\theta_1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ จะเป็น

เมทริกซ์ \mathbf{A} และเมทริกซ์ $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ จะแทนเมทริกซ์ \mathbf{B} ในสมการ state forecast ของ Kalman filter สมการที่ (2.20) และ (2.21) และเนื่องจากตัวแปรที่เราต้องการทำนายคือกำลังผลิตเหมือนกันกับตัวแปร state ดังนั้นเมทริกซ์ \mathbf{C} จะแทนด้วย $[1 \ 0 \ 0]$ จากนั้นเราจะให้ Kalman filter ทำนายหรือติดตามกำลังผลิตโดยที่ป้อนค่า \mathbf{y}_t ซึ่งคือค่ากำลังผลิตที่ได้จากการวัด

ในบทถัดไปจะเป็นการทดสอบวิธีการที่ได้นำเสนอโดยการทดลองให้ทำนายกำลังผลิตล่วงหน้า 1 วัน โดยแบ่งเป็น 2 กรณี กรณีแรกถ้าหากกำหนดให้ระบบทำงานตลอดเวลาและสถานี PV ยังสามารถเข้าถึงได้และส่งข้อมูลได้ทำให้ทราบกำลังผลิตแบบเวลาจริงของสถานีที่ต้องการทำนาย ดังนั้นจึงใช้กำลังผลิตที่วัดได้จริงแทนค่าใน \mathbf{y}_t ในกรณีที่สองเป็นกรณีที่สถานี PV ไม่สามารถส่งข้อมูลกำลังผลิตแบบเวลาจริงได้ดังนั้นเราจะไม่ทราบรายละเอียดกำลังผลิตของ PV และไม่สามารถติดตามกำลังผลิตได้ จึงใช้ Estimator model ในการประมาณค่ากำลังผลิตแบบเวลาจริงและแทนค่าใน \mathbf{y}_t ในสมการที่ (2.24) ของ Kalman filter เพื่อติดตามค่ากำลังผลิต

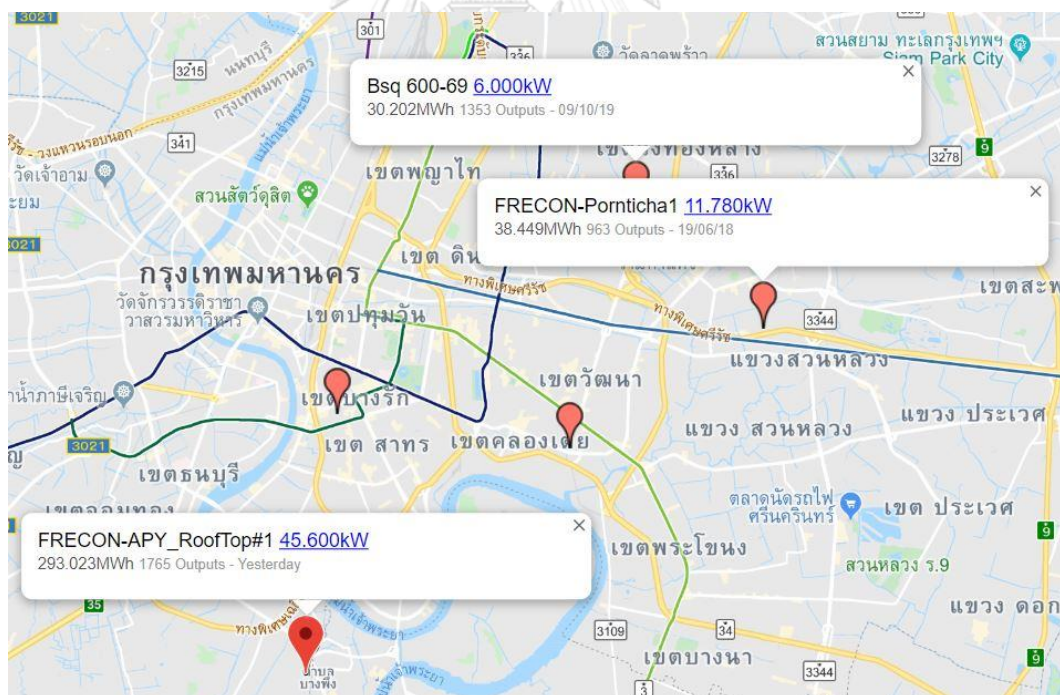
ในกรณีที่สองนั้นการทดสอบในบทถัดไปจะใช้ Persistence model และ Artificial Neural Network ในการเปรียบเทียบความสามารถในการทำนาย โดยการให้ทดสอบทำนายเป็นเวลา 1 วัน และกำหนดให้ PV ขาดการติดต่อในเวลาที่จะเริ่มวันใหม่พอดี เนื่องจากถ้าเราไม่ทราบกำลังผลิตทุก 5 นาทีของสถานีที่ต้องการทำนาย จะทำให้ไม่สามารถใช้ Persistence model หรือแบบจำลองอื่นทำนายทุก 5 นาทีได้เหมือนกับวิธีที่ใช้ Kalman filter ร่วมกับ Estimator model และการใช้ข้อมูลกำลังผลิตในอดีตจนถึงช่วงเวลาที่ PV ไม่สามารถส่งข้อมูลได้ fit กับ ARIMA Model ในการทำนายเพียงอย่างเดียวมีความแม่นยำที่น้อยกว่าการใช้ข้อมูลของค่าประมาณกำลังผลิตแบบเวลาจริงที่ได้จากการคำนวณของ Estimator model ช่วยในการทำนายผ่าน Kalman Filter ดังนั้นเราจึงใช้การทดสอบทำนาย 1 วันโดยใช้ข้อมูลล่าสุดที่เป็นไปได้ในการทำนาย รายละเอียดของแต่ละวิธีได้กล่าวไว้ในบทที่ 2

บทที่ 4

การทดสอบและผลลัพธ์

4.1 การทดสอบผลลัพธ์ของการเลือกใช้ชุดข้อมูลที่ต่างกันในการ fit ARIMA Model

เริ่มการทดสอบโดยสุ่มเลือก solar PV เพื่อทดสอบวิธีการที่เสนอจากตำแหน่งต่างกัน 3 ตำแหน่ง โดยทั้ง 3 สถานีมีชื่อเรียกดังนี้ Apy, Bsq และ Pomticha (รูปที่ 4-1) จากนั้นเก็บค่ากำลังผลิตและระยะทางระหว่าง solar PV แต่ละแห่งเพื่อใช้ในการคำนวณหาค่าประมาณจาก Estimator Model โดยค่ากำลังผลิตที่เก็บมาจะมีระยะห่างระหว่างข้อมูลเท่ากับ 5 นาทีและเก็บข้อมูลตั้งแต่วันที่ 7:00 น. ถึง 18:00 น. ในแต่ละวันเป็นเวลา 1 เดือน ในการทดสอบจะเลือก solar PV 1 แห่งเป็นสถานีที่ต้องการทำนายกำลังผลิตและจะใช้ solar PV ที่เหลือเพื่อคำนวณค่าประมาณกำลังผลิตแบบเวลาจริงจาก Estimator model ในกรณีนี้เราเลือก Apy เป็นสถานีที่ต้องการทำนายกำลังผลิต และกำหนดให้ทำนายกำลังผลิตของวันที่ 15/3/2018 เพื่อทดสอบในเบื้องต้น จากนั้นขยายผลการทดสอบโดยให้ทดสอบความแม่นยำกับข้อมูลหลายวันเพื่อดูพฤติกรรมของผลลัพธ์



รูปที่ 4-1 แผนที่แสดงที่ตั้งของ solar rooftop ทั้ง 3 แห่ง

วัตถุประสงค์การทดสอบ ต้องการศึกษาลักษณะของการเลือกใช้ชุดข้อมูลในการ fit ARIMA model และสภาพอากาศของวันที่ต้องการทดสอบมีผลอย่างไรต่อความแม่นยำในการทำนายกำลังผลิต (การทดสอบเลือกให้ทำนายวันที่ 15/3/2018 ของสถานี Apy ก่อนจากนั้นจึงทดสอบทุกวันเป็นเวลา 1 เดือน)

ข้อมูลที่ใช้ในการทดสอบ

- 1) กำลังผลิตของสถานีทั้ง 3 แห่งได้แก่ Apy, Bsq และ Pornticha โดยที่ข้อมูลกำลังผลิตมีทั้งหมด 30 วันและในแต่ละวันมีชุดข้อมูลจำนวน 133 ข้อมูลตั้งแต่เวลา 7:00 น. ถึง 18:00 น. และ ข้อมูลมีความละเอียดต่างกันทุก 5 นาที โดยเป็นข้อมูลกำลังผลิตที่ normalized แล้วตั้งสมการที่ (3.1)
- 2) ข้อมูลระยะทางระหว่างสถานีที่ต้องการทำนายกำลังผลิตกับสถานีข้างเคียงได้แก่ระยะทางระหว่างสถานี Apy กับ Bsq และ ระยะทางระหว่างสถานี Apy กับ Pornticha เท่านั้น

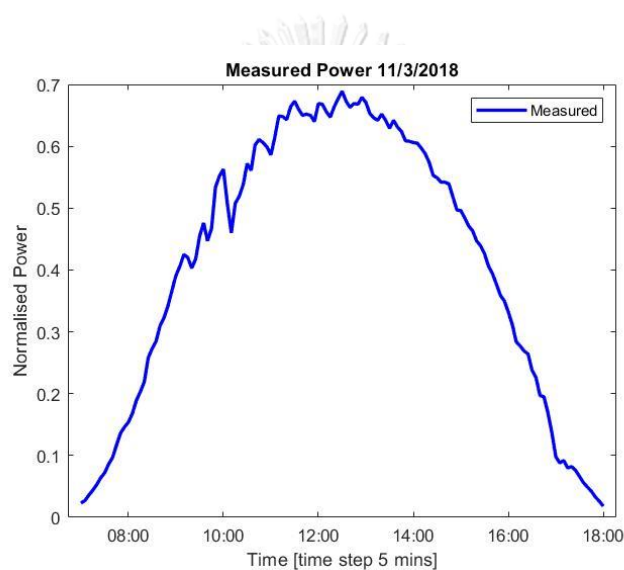
วิธีการทดสอบ

- 1) เลือกชุดข้อมูลที่จะใช้ fit ARIMA model ดังนี้ 1. ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศแจ่มใส 2. ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศมีเมฆมาก 3. ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศฝนตก
- 2) ใช้ ARIMA และ Kalman filter ทำนายกำลังผลิตในกรณีที่มีข้อมูลกำลังผลิตแบบเวลาจริง และ ในกรณีที่ไม่สามารถทราบกำลังผลิตแบบเวลาจริงของสถานีที่สนใจจะใช้ Estimator model ประมาณกำลังผลิตแบบเวลาจริงก่อนแล้วจึงใช้ Kalman filter
- 3) คำนวณค่าประมาณกำลังผลิตของสถานี Apy จากสมการ Estimator model โดยใช้กำลังผลิตแบบเวลาจริงของสถานี Bsq และ Pornticha ในวันเดียวกับวันที่ต้องการทำนาย
- 4) ทำนายและติดตามกำลังผลิตของสถานี Apy โดยใช้ Kalman filter algorithm และคำนวณดัชนีชี้วัดเพื่อดูความแม่นยำของการทำนาย

ผลลัพธ์ของการทดสอบ

4.1.1 กรณีใช้ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศแจ่มใส

ในการทดสอบนี้ใช้ข้อมูลของวันที่มีสภาพอากาศแจ่มใสโดยสุ่มเลือกวันที่มีลักษณะอากาศแจ่มใสจากข้อมูลกำลังผลิตที่มีดังรูปที่ 4-2 จากนั้นนำข้อมูลกำลังผลิตของวันที่สุ่มมานี้สร้าง ARIMA model และทำนายกำลังผลิตโดยใช้ Kalman filter ผลลัพธ์ของการทำนายนำไปคำนวณดัชนีชี้วัดได้ผลลัพธ์ดังตารางที่ 4-1 โดยคอลัมน์ที่สามของตารางเป็นกรณีที่ใช้ Estimator model ช่วยประมาณค่ากำลังผลิตแบบเวลาจริงก่อนทำนายโดย Kalman filter



รูปที่ 4-2 ตัวอย่างรูปกำลังผลิตในวันที่ท้องฟ้าแจ่มใส

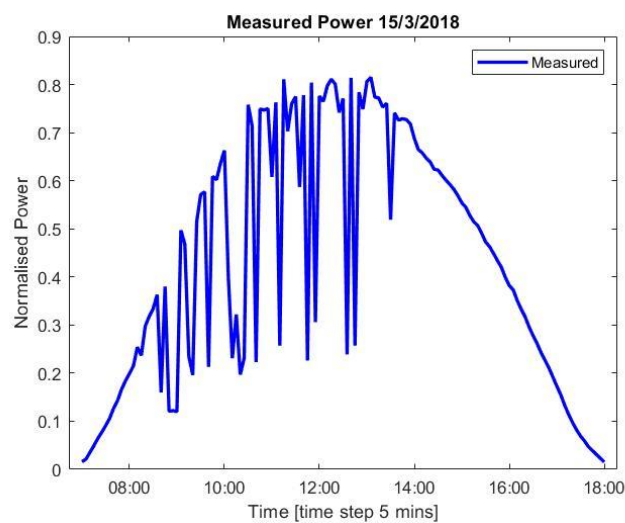
CHULALONGKORN UNIVERSITY

ตารางที่ 4-1 ตารางชี้วัดความแม่นยำในกรณีใช้ข้อมูลวันที่อากาศแจ่มใสในการสร้างแบบจำลองเพื่อทำนายกำลังผลิตในวันที่ 15/3/2018

Index	Kalman Filter with real-time data	Kalman Filter with Estimator model
RMSE	0.0578	0.1321
nRMSE	0.1333	0.3049
MSE	0.0033	0.0174
MAPE	9.1400	27.5752

4.1.2 กรณีใช้ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศมีเมฆมาก

ในการทดสอบนี้ใช้ข้อมูลของวันที่มีสภาพอากาศมีเมฆมากโดยสุ่มเลือกวันที่มีลักษณะอากาศมีเมฆมากจากข้อมูลกำลังผลิตที่มีดังรูปที่ 4-3 จากนั้นนำข้อมูลกำลังผลิตของวันที่สุ่มมานี้สร้าง ARIMA model และทำนายกำลังผลิตโดยใช้ Kalman filter ผลลัพธ์ของการทำนายนำไปคำนวณดัชนีชี้วัดได้ผลลัพธ์ดังตารางที่ 4-2 โดยคอลัมน์ที่สามของตารางจะเป็นกรณีที่ใช้ Estimator model ช่วยประมาณค่ากำลังผลิตแบบเวลาจริงก่อนทำนายโดย Kalman filter



รูปที่ 4-3 รูปกำลังผลิตขณะสภาพอากาศมีเมฆมาก

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

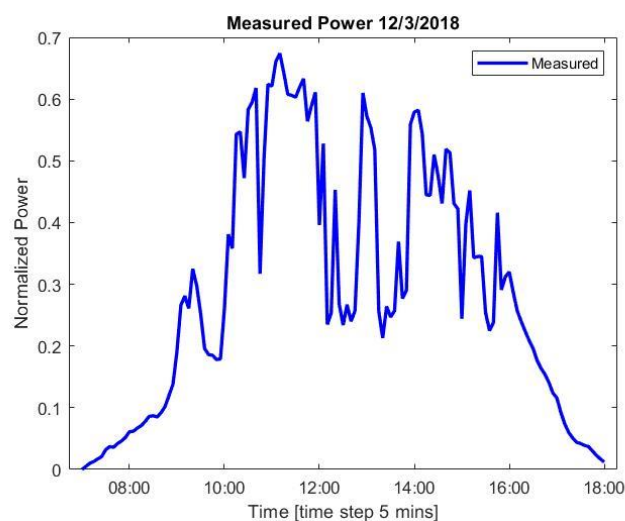
ตารางที่ 4-2 ตารางชี้วัดความแม่นยำในกรณีใช้ข้อมูลวันที่อากาศมีเมฆมาก

ในการสร้างแบบจำลองเพื่อทำนายกำลังผลิตในวันที่ 15/3/2018

Index	Kalman Filter with real-time data	Kalman Filter with Estimator model
RMSE	0.0650	0.1329
nRMSE	0.1500	0.3069
MSE	0.0042	0.0177
MAPE	18.6784	35.7741

4.1.3 กรณีใช้ชุดข้อมูลที่มีกราฟกำลังผลิตที่มีลักษณะสภาพอากาศฝนตก

ในการทดสอบนี้ใช้ข้อมูลของวันที่มีสภาพอากาศฝนตกโดยสุ่มเลือกวันที่มีลักษณะอากาศฝนตกจากข้อมูลกำลังผลิตที่มีดังรูปที่ 4-4 จากนั้นนำข้อมูลกำลังผลิตของวันที่สุ่มมานี้สร้าง ARIMA model และทำนายกำลังผลิตโดยใช้ Kalman filter ผลลัพธ์ของการทำนายนำไปคำนวณดัชนีชี้วัดได้ผลลัพธ์ดังตารางที่ 4-3 โดยคอลัมน์ที่สามของตารางจะเป็นกรณีที่ใช้ Estimator model ช่วยประมาณค่ากำลังผลิตแบบเวลาจริงก่อนทำนายโดย Kalman filter



รูปที่ 4-4 รูปกำลังผลิตขณะสภาพอากาศมีฝนตก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

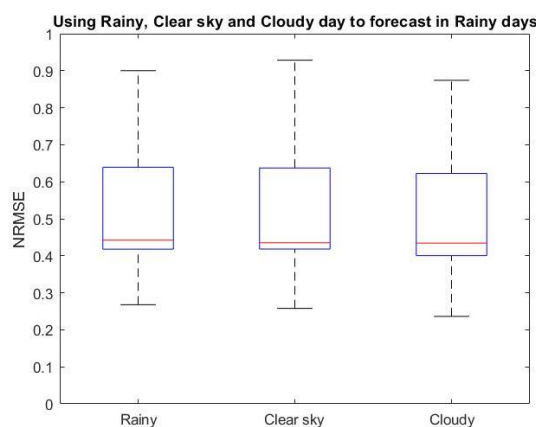
ตารางที่ 4-3 ตารางชี้วัดความแม่นยำในกรณีใช้ข้อมูลวันที่มีลักษณะสภาพอากาศฝนตก

ในการสร้างแบบจำลองเพื่อทำนายกำลังผลิตในวันที่ 15/3/2018

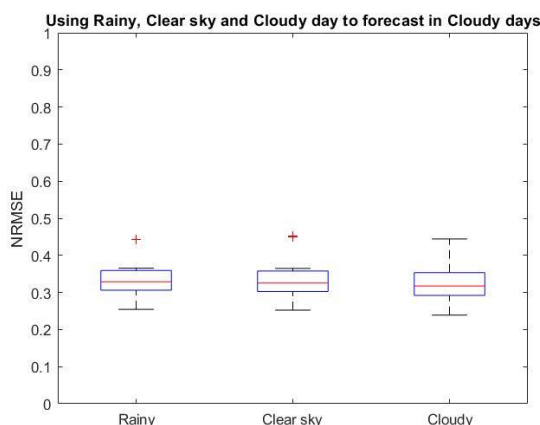
Index	Kalman Filter with real-time data	Kalman Filter with Estimator model
RMSE	0.0627	0.1340
nRMSE	0.1447	0.3093
MSE	0.0039	0.0179
MAPE	13.8567	31.5632

4.1.4 กรณีที่ใช้วันทั้ง 3 แบบข้างต้นเพื่อทำนายกำลังผลิตเป็นเวลา 1 เดือน

ในการทดลองนี้เป็นการขยายผลการทดลองโดยนอกจากจะทำนายในวันที่ 15/3/2018 แล้ว จะใช้วันทั้ง 3 แบบที่ได้สุ่มมาจากการทดลองก่อนหน้านี้ทำนายกำลังผลิตของทุกวันเป็นเวลา 1 เดือน เพื่อดูพฤติกรรมของผลลัพธ์ โดยจะแบ่งรูปแบบสภาพอากาศของวันที่ทำนายในเดือนที่เก็บข้อมูลได้ 2 รูปแบบคือ 1. สภาพอากาศที่มีฝนตก และ 2. สภาพอากาศที่มีเมฆมาก รูปที่ 4-5 เป็นรูปที่แสดงผลลัพธ์ของค่า NRMSE เมื่อใช้ข้อมูลวันที่มีสภาพอากาศฝนตก, แจ่มใส และมีเมฆมากในการ fit ARIMA Model แล้วทำนายกำลังผลิตของวันที่มีสภาพอากาศฝนตก และ รูปที่ 4-6 จะเป็นรูปที่แสดงผลลัพธ์ของค่า nRMSE เมื่อใช้วันที่มีสภาพอากาศฝนตก, แจ่มใส และมีเมฆมากในการ fit ARIMA Model แล้วทำนายกำลังผลิตของวันที่มีสภาพอากาศมีเมฆมาก ขอบด้านบนและด้านล่างของกล่องแต่ละใบ ในรูปคือค่า Percentile ที่ 25 และ 75 ของชุดข้อมูล ซีดกลางกล่องคือค่า Median ของชุดข้อมูล เส้นประที่ลากจากขอบบนและขอบล่างของกล่องไปยังด้านบนและด้านล่างบ่งบอกถึงจุดสูงสุดและจุดต่ำสุดของข้อมูลและขนาดของกล่องในแนวตั้งบ่งบอกถึงส่วนเบี่ยงเบนมาตรฐานของแต่ละชุดข้อมูล



รูปที่ 4-5 ผลลัพธ์ของค่า nRMSE เมื่อใช้วันที่มีสภาพอากาศฝนตก, แจ่มใส และมีเมฆมากทำนายกำลังผลิตของวันที่มีสภาพอากาศฝนตก



รูปที่ 4-6 ผลลัพธ์ของค่า $nRMSE$ เมื่อใช้วันที่มีสภาพอากาศฝนตก, แจ่มใส และ มีเมฆมากทำนายกำลังผลิตของวันที่มีสภาพอากาศมีเมฆมาก

4.1.5 สรุปผลการทดสอบ

จากการทดสอบได้ผลลัพธ์ตามตารางที่ 4-1 ตารางที่ 4-2 และตารางที่ 4-3 ตามลำดับ ในการทดสอบเราได้ใช้ข้อมูลวันที่มีลักษณะของสภาพอากาศแตกต่างกันในการสร้างแบบจำลอง ARIMA model ที่ต่างกันเพื่อประมาณค่า coefficient ในการใช้ Kalman filter ทำนายกำลังผลิต พบว่าหลังจากที่สร้างแบบจำลอง ARIMA แล้วใช้ Kalman filter ทำนายกำลังผลิต ผลลัพธ์จากดัชนีชี้วัดในแต่ละกรณีใกล้เคียงกัน แต่ถ้าใช้ข้อมูลวันที่มีสภาพอากาศแจ่มใสในการ fit แบบจำลองจะได้ผลลัพธ์จากการทำนายที่ดีกว่ากรณีอื่นเล็กน้อย แสดงให้เห็นว่า Kalman filter กับ Estimator model สามารถทำนายและติดตามกำลังผลิตแบบเวลาจริงได้อย่างมีประสิทธิภาพเนื่องจาก Estimator model สามารถคำนวณค่าประมาณของกำลังผลิตแบบเวลาจริงจาก PV ข้างเคียงเพื่อกำหนดทิศทางของกราฟกำลังผลิต นอกจากนั้นเมื่อขยายผลการทดลองโดยใช้วันที่มีสภาพอากาศแตกต่างกัน 3 แบบข้างต้นทำนายกำลังผลิตของวันอื่นๆทุกวันเป็นเวลา 1 เดือนได้ผลลัพธ์ดังรูปที่ 4-5 และ 4-6 พบว่าทั้ง 3 วันให้ผลลัพธ์ค่า $nRMSE$ ใกล้เคียงกันสอดคล้องกับการทดลองเบื้องต้น นอกจากนั้นยังพบว่าเมื่อวันที่ต้องการทำนายกำลังผลิตมีสภาพอากาศฝนตกจะทำให้ภาพรวมผลลัพธ์ของการทำนายต่ำกว่าเมื่อเปรียบเทียบกับวันที่มีสภาพอากาศมีเมฆมากเนื่องจากผลลัพธ์ที่ได้มีความคลาดเคลื่อนมากสาเหตุมาจากสภาพอากาศแบบฝนตกลักษณะของกำลังผลิตจะแตกต่างกันมากกว่าสภาพอากาศแบบมีเมฆมากทำให้มีโอกาสที่จะคลาดเคลื่อนมากกว่า

4.2 Case study 1: 3 Solar PV System

วัตถุประสงค์การทดสอบ ต้องการทดสอบวิธีการทำนายที่เสนอโดยสมมติให้ทำนายกำลังผลิตในวันใดวันหนึ่งภายในชุดข้อมูลที่เก็บมาและแสดงขั้นตอนในการทำนายกำลังผลิต (การทดสอบสมมติให้ทำนายวันที่ 15/3/2018 ของสถานี Apy) นอกจากนี้ยังเปรียบเทียบวิธีการที่นำเสนอกับการทำนายจาก neural network และ persistence model

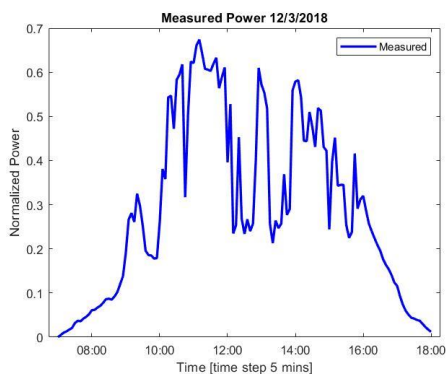
ข้อมูลที่ใช้ในการทดสอบ 1) กำลังผลิตของสถานีทั้ง 3 แห่งได้แก่ Apy, Bsq และ Pornticha โดยที่ข้อมูลกำลังผลิตมีทั้งหมด 30 วันและในแต่ละวันมีชุดข้อมูลจำนวน 133 ข้อมูลตั้งแต่เวลา 7:00 น. ถึง 18:00 น. และ ข้อมูลมีความละเอียดต่างกันทุก 5 นาที โดยเป็นข้อมูลกำลังผลิตที่ normalized แล้วดังสมการที่ (3.1)

2) ข้อมูลระยะทางระหว่างสถานี Apy กับ Bsq และ ระยะทางระหว่างสถานี Apy กับ Pornticha (เนื่องจากเราต้องการทำนายกำลังผลิตของ Apy ดังนั้นตัวแปรระยะทางที่จำเป็นคือระยะทางระหว่างสถานี Apy กับ Bsq และ ระยะทางระหว่างสถานี Apy กับ Pornticha เท่านั้น)

วิธีการทดสอบและผลลัพธ์ของการทดสอบ

4.2.1 Data Inspection

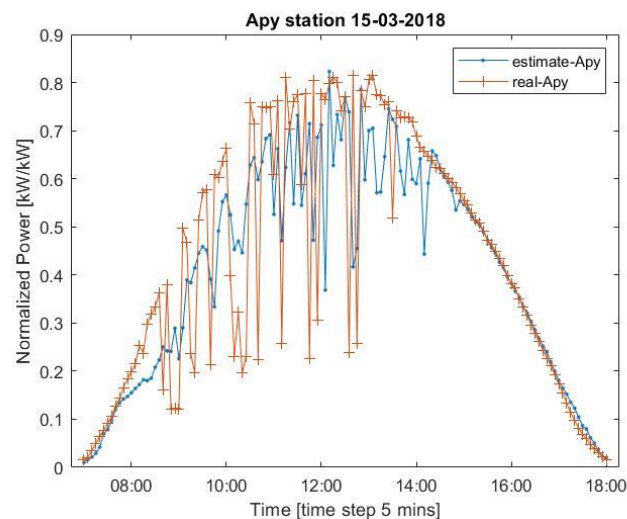
นำข้อมูลกำลังผลิตของ solar PV ทั้ง 3 แห่งมาตรวจสอบว่ามีข้อมูลที่หายไปหรือไม่ ถ้ามีข้อมูลที่หายไปบางจุดให้ใช้การประมาณค่าเชิงเส้นในการประมาณค่าข้อมูลที่หายไป รูปที่ 4-7 แสดงตัวอย่างกำลังผลิตของสถานี Apy ในวันที่ 12/3/2018 ต่อไปเราจะใช้กำลังผลิตในวันที่กล่าวถึงนี้ในการวิเคราะห์และสร้างแบบจำลอง ARIMA model



รูปที่ 4-7 กำลังผลิตของสถานี Apy ในวันที่ 12/3/2018

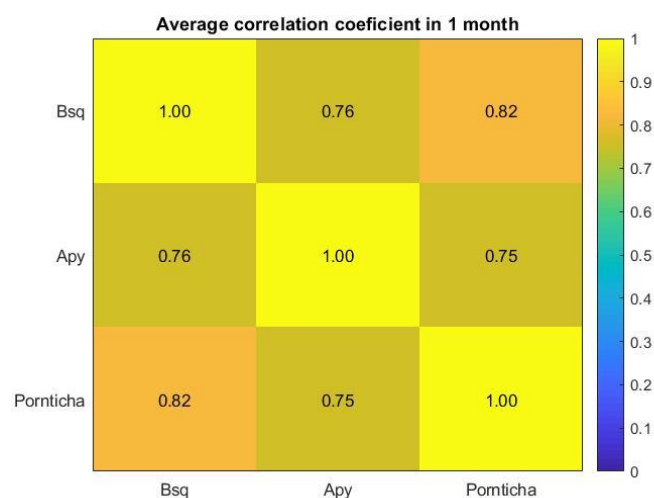
4.2.2 Calculating Power Generation by using Estimator Model

ในขั้นตอนนี้เราจะคำนวณกำลังผลิตแบบเวลาจริงของสถานี Apy เพื่อใช้ใน Kalman Filter algorithm ในกรณีที่เราไม่สามารถทราบกำลังผลิตแบบเวลาจริงของสถานีที่เราต้องการที่จะทำนาย โดยสามารถคำนวณได้จากสมการที่ (3.2) ที่ได้กล่าวไปแล้วข้างต้นในหัวข้อ การออกแบบ Estimator model สมการข้างต้นใช้ข้อมูลกำลังผลิตแบบเวลาจริงในวันที่ 15/3/2018 ของสถานี Bsq และ Pornticha เพื่อคำนวณค่าประมาณกำลังผลิตแบบเวลาจริงของ Apy ผลลัพธ์ที่ได้จากการคำนวณ แสดงได้ดังรูปที่ 4-8 นอกจากนั้นเมื่อตรวจสอบค่า correlation coefficient ของทั้ง 3 สถานีในช่วง เวลา 1 เดือนพบว่าค่าเฉลี่ยของ correlation ระหว่างสถานีทั้ง 3 มีค่าเป็นบวกและมีค่าอยู่ในช่วง 0.6 ถึง 1 ดังรูปที่ 4-9 ทำให้สามารถสรุปได้ว่าค่าประมาณจาก Estimator model มีความเชื่อถือได้



รูปที่ 4-8 รูปภาพเปรียบเทียบค่าประมาณที่ได้จากสมการของ

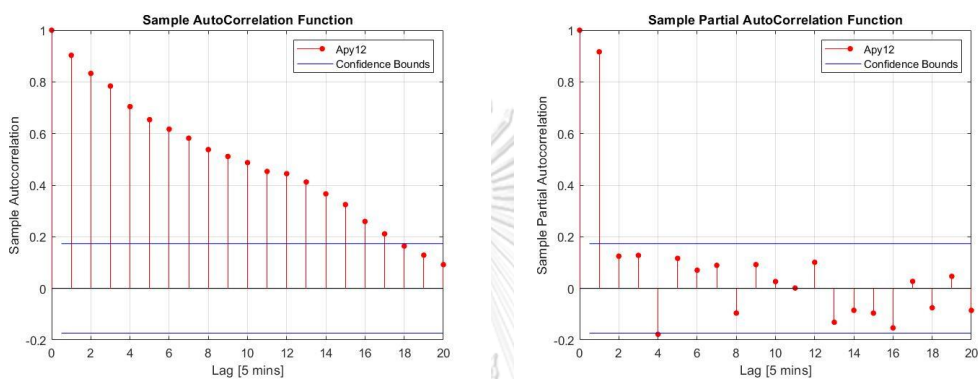
Estimator model กับค่าจริง



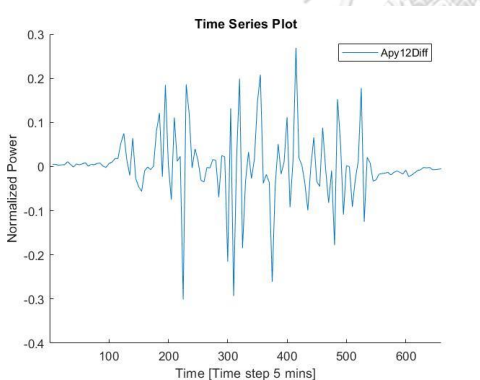
รูปที่ 4-9 ค่าเฉลี่ยของ correlation ในเวลา 1 เดือน

4.2.3 Created ARIMA Model

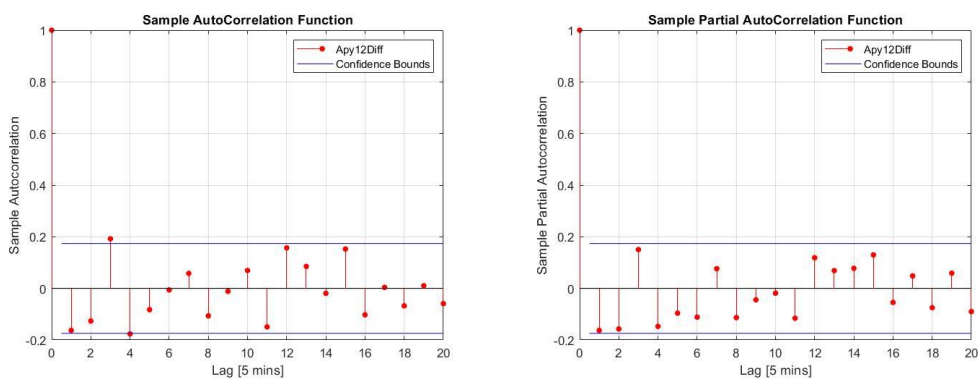
จากข้อมูลกำลังผลิตของสถานี Apy ในวันที่ 12/3/2018 เรานำข้อมูลมาวิเคราะห์เพื่อสร้าง ARIMA model พบว่าข้อมูลมีความเป็น non-stationary time series จึง plot ACF และ PACF เพื่อตรวจสอบ จากนั้นทำ differencing ข้อมูลได้ผลลัพธ์ดังรูปที่ 4-10 และ 4-11 หลังจากทำ differencing ข้อมูลแล้วพบว่าข้อมูลมีความเป็น stationary มากขึ้นจึงตรวจสอบ ACF และ PACF อีกครั้งได้ผลลัพธ์ดังรูปที่ 4-12 จากนั้นใช้ตารางที่ 2-1 เพื่อเลือกระบบค่า p และ q ในเบื้องต้น



รูปที่ 4-10 ACF plot และ PACF plot ของกำลังผลิตสถานี Apy



รูปที่ 4-11 กำลังผลิตหลังจาก difference 1 ครั้งของ Apy ในวันที่ 12/3/2018

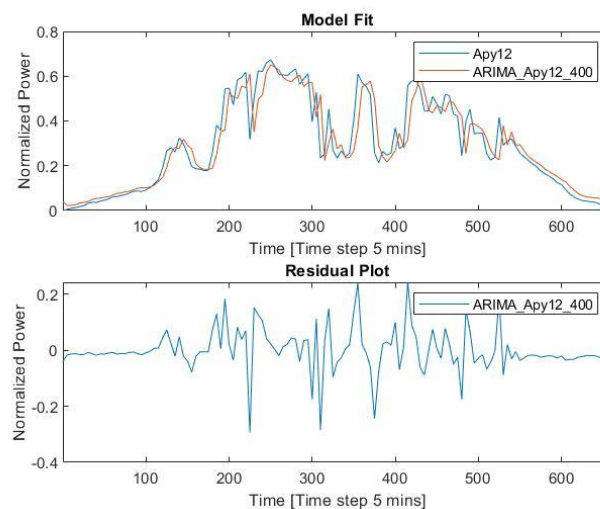


รูปที่ 4-12 ACF plot และ PACF plot ของกำลังผลิตสถานี Apy หลังจาก difference 1 ครั้ง

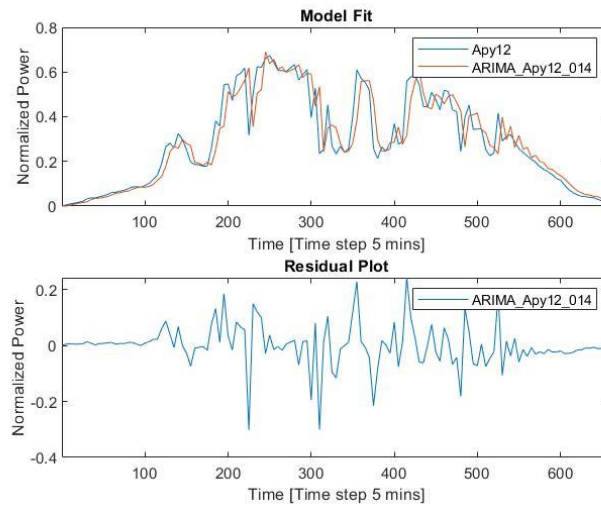
จากผลลัพธ์ข้างต้นเราคาดเดาว่าน่าจะเป็น ARIMA(4,0,0) หรือ ARIMA(0,1,4) เมื่อระบุค่า p q และ d ได้แล้วเราใช้ MATLAB ในการประมาณค่าสัมประสิทธิ์ของ ARIMA model โดย MATLAB ใช้หลักการของ maximum likelihood method ผลลัพธ์ที่ได้เป็นไปดังรูปที่ 4-15 และรูปที่ 4-16 แสดงค่าสัมประสิทธิ์ของ AR และ MA โดยที่ AR{1}, AR{2}, AR{3}, ... แทนค่าสัมประสิทธิ์ของ AR ที่ตำแหน่งต่างกันในสมการ ARIMA และ MA{1}, MA{2}, MA{3}, ... แทนค่าสัมประสิทธิ์ของ MA ที่ตำแหน่งต่างกันในสมการ ARIMA ตัวแปร constant คือค่าคงที่ของแบบจำลองและตัวแปร variance คือค่า variance ของแบบจำลอง คอลัมน์ Standard Error คือค่าประมาณของ standard deviation ของ Error บ่งบอกถึงความแปรปรวนของ Error และค่า t-statistic เกิดจากการนำคอลัมน์ value หารด้วย Standard Error และคอลัมน์ P-value ใช้ในการตรวจสอบว่าค่า value ในแถวเดียวกันมีนัยสำคัญต่อแบบจำลองหรือไม่ โดยถ้า P-value มีค่าน้อยกว่า 0.05 แสดงว่าค่า value ที่อยู่ในแถวเดียวกันมีนัยสำคัญ ถ้าหากค่า P-value มากกว่า 0.05 เราอาจพิจารณาได้ว่าค่า value ที่สอดคล้องกับค่า P-value นั้นมีค่าเท่ากับ 0

จากผลลัพธ์พบว่าค่า residual histogram และ ACF plot ของ ARIMA(0,1,4) เหมาะสมกว่าเนื่องจากกราฟกำลังผลิตรมีความเป็น stationary มากขึ้นดูได้จากกราฟ ACF ที่ลดตัวอย่างรวดเร็ว และค่าเฉลี่ยคงที่ หลังจากทำ differencing 1 ครั้งในรูปที่ 4-12 นอกจากนั้น residual histogram ในรูปที่ 4-18 มีลักษณะ normally distributed และ ACF plot มีลักษณะ uncorrelated ดังนั้นเราจะใช้ ARIMA(0,1,4) ในการสร้าง state-space model เพื่อคำนวณ Kalman Filter

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4-13 Residual plot ของ ARIMA(4,0,0)



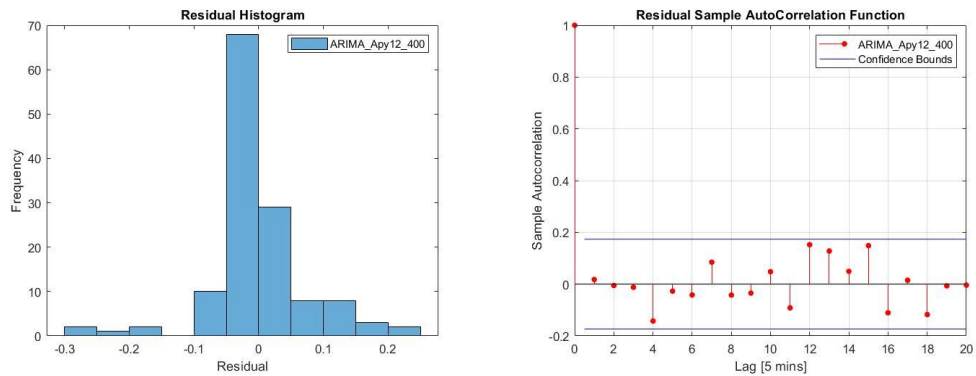
รูปที่ 4-14 Residual plot ของ ARIMA(0,1,4)

Parameters				
Parameter	Value	Standard Error	t Statistic	P-Value
Constant	0.0204	0.0312	0.6557	0.5120
AR{1}	0.8117	0.0801	10.1299	4.0718e-24
AR{2}	0.0266	0.0802	0.3314	0.7404
AR{3}	0.2684	0.0913	2.9396	0.0033
AR{4}	-0.1778	0.0809	-2.1988	0.0279
Variance	0.0059	6.6252e-04	8.9136	4.9413e-19

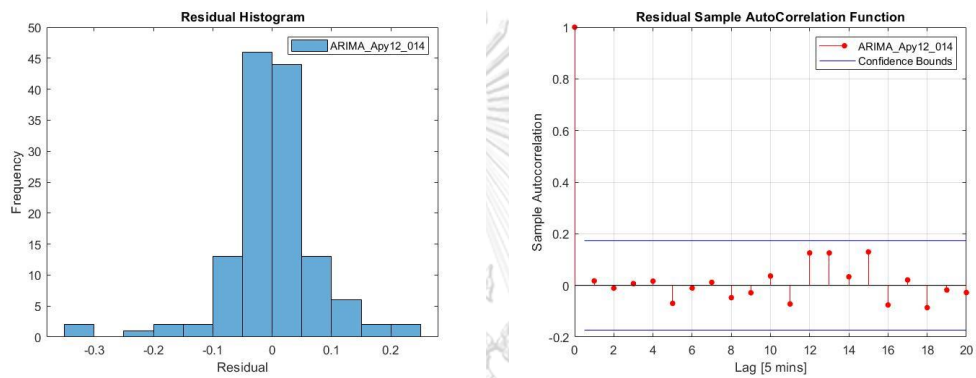
รูปที่ 4-15 parameter ของ ARIMA(4,0,0)

Parameters				
Parameter	Value	Standard Error	t Statistic	P-Value
Constant	1.3971e-04	0.0044	0.0321	0.9744
MA{1}	-0.1770	0.0776	-2.2805	0.0226
MA{2}	-0.1155	0.0658	-1.7557	0.0791
MA{3}	0.1670	0.0857	1.9482	0.0514
MA{4}	-0.2498	0.0784	-3.1875	0.0014
Variance	0.0058	4.6079e-04	12.6624	9.5497e-37

รูปที่ 4-16 paramater ของ ARIMA(0,1,4)



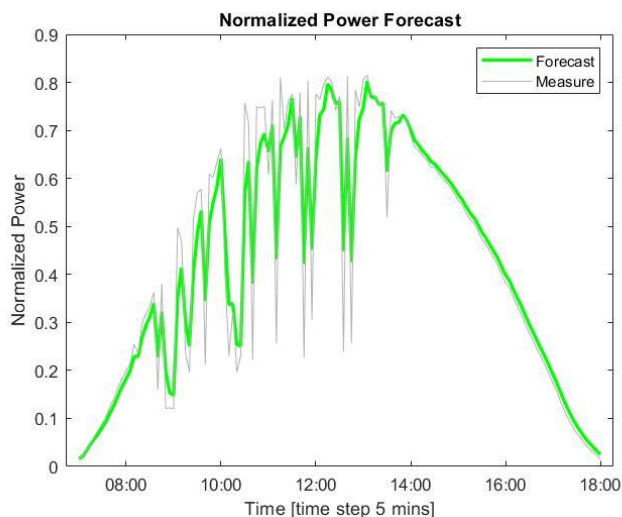
รูปที่ 4-17 Residual Histogram และ ACF plot ของ ARIMA(4,0,0)



รูปที่ 4-18 Residual Histogram และ ACF plot ของ ARIMA(0,1,4)

4.2.4 Forecast without Estimator Model (Observation data is available)

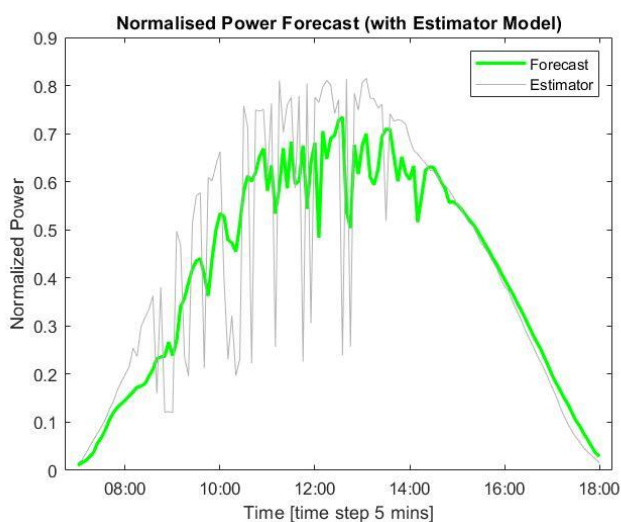
ในกรณีที่สามารถทราบค่ากำลังผลิตแบบเวลาจริงของสถานี Apy ได้เราจะใช้ค่านั้นใน Kalman Filter algorithm ในขั้นตอนนี้เราจะเขียน ARIMA model ในรูปของ state-space model จากนั้นเราจะใช้ coefficient matrix ของ ARIMA model ในสมการ state forecast ของ Kalman Filter รูปที่ 4-19 แสดงผลลัพธ์ของการทำนายจาก Kalman filter เทียบกับค่าจริง



รูปที่ 4-19 กำลังผลิตของสถานี Apy ในวันที่ 15/3/2018 จากการทำนายที่เป็นผลลัพธ์จาก Kalman filter โดยใช้ข้อมูลกำลังผลิตแบบเวลาจริง

4.2.5 Forecast with Estimator Model (Observation data is unavailable)

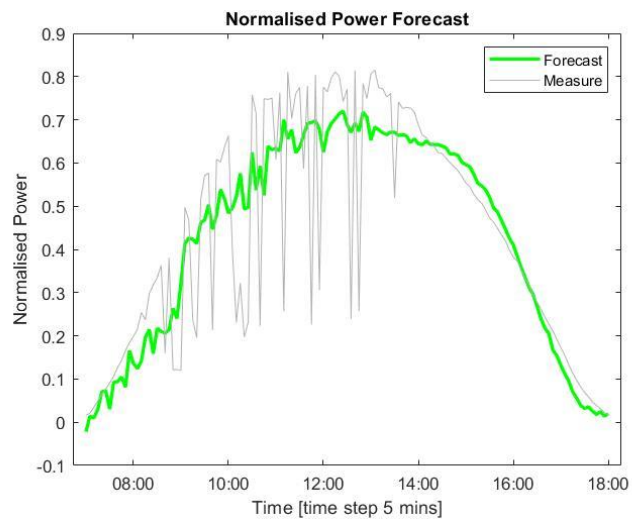
ในกรณีที่ไม่สามารถทราบค่ากำลังผลิตแบบเวลาจริง จะใช้ Estimator model เพื่อคำนวณค่ากำลังผลิตแบบเวลาจริงก่อนแล้วจึงใช้ Kalman filter algorithm เพื่อทำนายกำลังผลิต รูปที่ 4-20 แสดงผลลัพธ์ของการทำนายจาก kalman filter เทียบกับค่าจริง



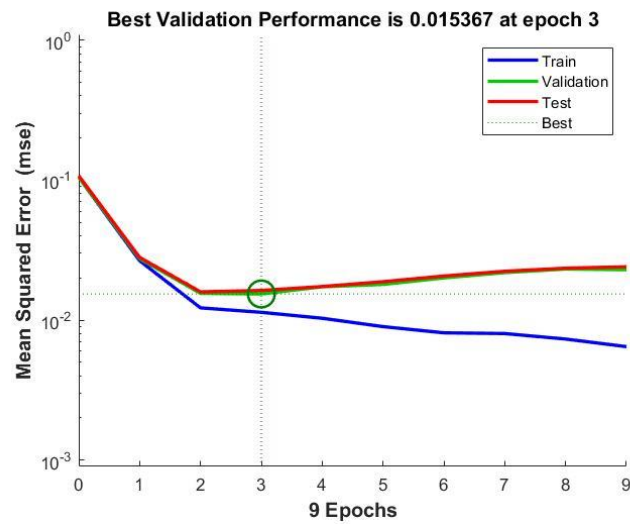
รูปที่ 4-20 กำลังผลิตของสถานี Apy ในวันที่ 15/3/2018 จากการทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator model คำนวณค่าประมาณข้อมูลกำลังผลิตแบบเวลาจริง

4.2.6 Forecasting by Artificial Neural Network

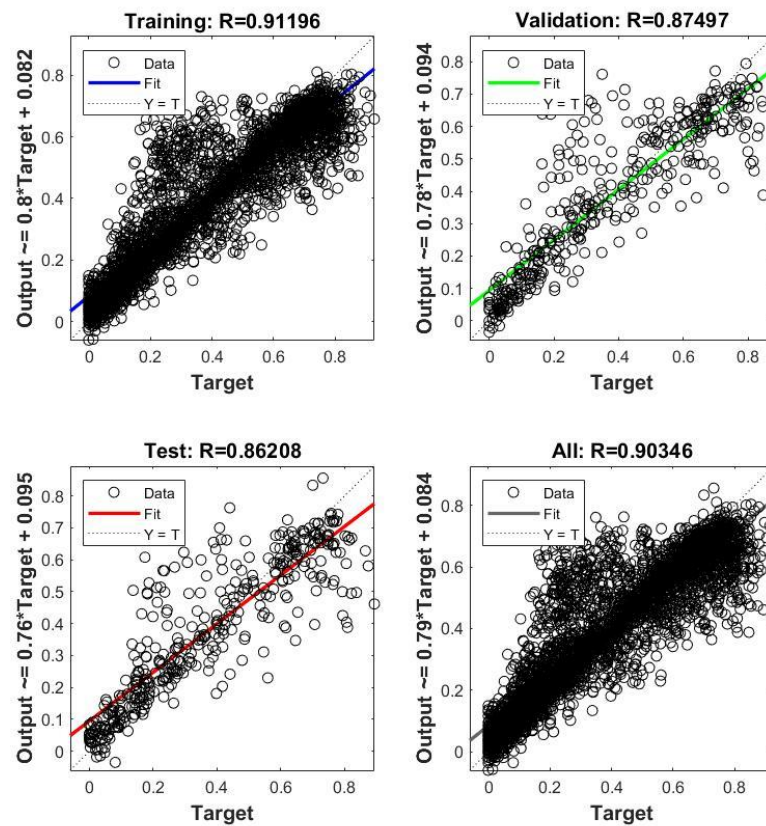
ในหัวข้อนี้เราจะทำนายกำลังผลิตล่วงหน้า 1 วันโดยใช้วิธีการทำนายที่มีพื้นฐานจาก Artificial Neural Network ตามขั้นตอนที่ได้แสดงให้เห็นในบทที่ 2 หัวข้อ Artificial Neural Network โดยใช้อัลกอริทึมแบบ Levenberg-Marquardt ในการ train แบบจำลองของสถานี Apy โดยใช้ข้อมูลกำลังผลิตของสถานี Apy ทั้ง 31 วันเป็นข้อมูลขาเข้าไปยัง neural network ที่มีจำนวน neuron ใน hidden layer เท่ากับ 10 (ในขั้นตอนนี้สามารถทดลอง train แบบจำลองในรูปแบบต่างๆตามที่กล่าวไว้ในบทที่ 2 และเนื่องจากข้อมูลกำลังผลิตที่จำนวนไม่มากสามารถใช้จำนวน hidden layer เท่ากับ 10 เพียงพอต่อการ train แบบจำลอง) หลังจากนั้นได้ผลลัพธ์ดังรูปที่ 4-21, 4-22 และ 4-23 ผลลัพธ์ระหว่างการทำนายจาก neural network เปรียบเทียบกับค่าจริงจากการวัด แสดงให้เห็นดังรูปที่ 4-21 ในส่วนของค่าจากกราฟ performance หรือค่า MSE ในรูปที่ 4-22 แสดงให้เห็นว่าแบบจำลองที่ train ให้ผลลัพธ์ค่า MSE ที่ดีดูจากเส้นกราฟ Test และ Validation มีความใกล้เคียงกันมาก นอกจากนั้นกราฟ regression ในรูปที่ 4-23 ยังมีผลลัพธ์ค่า R (ค่า correlation ระหว่างชุดข้อมูลขาออกกับค่าเป้าหมาย) สูงถึง 0.9 (เข้าใกล้ 1) ในกรณีของ Test set และ Validation set แสดงให้เห็นว่าค่าผลลัพธ์ของข้อมูลขาออกเข้าใกล้กับค่าเป้าหมายมาก



รูปที่ 4-21 กำลังผลิตของสถานี Apy ในวันที่ 15/3/2018 จาก
การทำนายโดยใช้ Artificial Neural Network เปรียบเทียบกับ
ค่าจริงที่ได้จากการวัด



รูปที่ 4-22 กราฟค่า performance เปรียบเทียบกันระหว่าง
ชุดข้อมูลที่ใช้ในการ train แบบจำลองของสถานี Apy



รูปที่ 4-23 กราฟ regression เปรียบเทียบกันระหว่างชุดข้อมูลที่ใช้ในการ train แบบจำลองของสถานี Apy

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

ผลลัพธ์จากในรูปที่ 4-23 เกิดจากการคำนวณค่า correlation ระหว่างผลลัพธ์จากการใช้ชุดข้อมูลต่างๆกับชุดข้อมูลเป้าหมาย โดยที่รูปทางด้านซ้ายบนคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูล training กับชุดข้อมูลเป้าหมาย รูปทางด้านขวาบนคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูล validation กับชุดข้อมูลเป้าหมาย รูปทางด้านซ้ายล่างคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูล test กับชุดข้อมูลเป้าหมาย และ รูปทางด้านขวาล่างคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูลทั้งหมดกับชุดข้อมูลเป้าหมาย โดยที่ชุดข้อมูล training, validation และ test เราได้แบ่งออกมาจากชุดข้อมูลรวมทั้งหมดในขั้นตอน dividing data set ที่ได้กล่าวไปในบทที่ 2 หัวข้อ Artificial Neural Network

4.2.7 Forecasting Result

ในขั้นตอนทำนายที่ผ่านมาเราได้ทำนายกำลังผลิตของสถานี Apy ในวันที่ 15/3/2018 และได้นำผลลัพธ์จากการทำนายไปเปรียบเทียบกับค่าจริงที่ได้เก็บไว้ล่วงหน้า ขั้นตอนต่อไปเราจะวัดค่าความแม่นยำของแต่ละกรณีออกมาเป็นตัวเลขเพื่อความชัดเจน โดยใช้สมการในหัวข้อ การทดสอบค่าความแม่นยำของผลลัพธ์จากแบบจำลองได้แก่ สมการที่ (2.26) ถึง สมการที่ (2.32) ในการคำนวณโดยเลือกใช้ค่า RMSE, nRMSE, MSE, skill score และ MAPE ผลลัพธ์จากการคำนวณความแม่นยำได้ตามตารางที่ 4-4

ตารางที่ 4-4 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิตของสถานี Apy ในวันที่ 15/3/2018

Index	Persistence model	Artificial Neural Network	Kalman Filter with Estimator model	Kalman Filter with real-time data
RMSE	0.1903	0.1317	0.1330	0.0626
nRMSE	0.4394	0.3041	0.3070	0.1446
MSE	0.0362	0.0173	0.0177	0.0039
Skill Score	-	0.5377	0.5119	0.8917
MAPE	29.8969	33.8301	31.8869	13.7414

4.2.8 สรุปผลการทดสอบ

จากตารางที่ 4-4 พบว่าผลลัพธ์จากการทำนายโดยใช้ Kalman Filter ร่วมกับกำลังผลิตแบบเวลาจริงของสถานีที่ต้องการทำนาย มีความแม่นยำสูงที่สุดโดยมีผลลัพธ์ค่า RMSE, nRMSE MSE และ MAPE ต่ำที่สุด ตามที่ได้คาดการณ์เอาไว้เนื่องจากจุดประสงค์ของการทดสอบกรณีนี้คือ การที่เมื่อปล่อยให้ Kalman filter ทำงานตลอดเวลาทั้งในกรณีที่ทราบกำลังผลิตอยู่แล้วเพื่อความต่อเนื่องในการทำงานของ Kalman filter เมื่อเกิดกรณีที่ไม่สามารถทราบกำลังผลิตแบบเวลาจริงจึงค่อยเปลี่ยนมาใช้กำลังผลิตที่ประมาณค่าได้จาก Estimator model ดังนั้นค่าผลลัพธ์ที่ได้จากกรณีนี้ควรใกล้เคียงกับของจริงให้มากที่สุด

นอกจากนี้กรณีที่มีความแม่นยำที่ต่ำสุดถัดมาคือกรณีที่ใช้ Estimator model คำนวณกำลังผลิตแบบเวลาจริงก่อนแล้วจึงใช้ค่านั้นร่วมกับ Kalman Filter ซึ่งการทดลองนี้แสดงให้เห็นว่า Estimator model สามารถใช้ประมาณค่ากำลังผลิตแบบเวลาจริงและนำมาใช้กับ Kalman Filter ได้โดยที่ผลลัพธ์อยู่ในระดับที่ดี เห็นได้จากกราฟผลลัพธ์ในรูปที่ 4-20 เมื่อเทียบกับรูปที่ 4-19 ซึ่งใช้ข้อมูลกำลังผลิตของจริงนอกจากนั้นยังแสดงให้เห็นถึงการเคลื่อนที่ของเส้นกราฟตามเส้นกราฟกำลังผลิตที่แท้จริงอยู่ นอกจากนี้เมื่อเปรียบเทียบกับการทำนายแบบที่ใช้ Artificial Neural Network แล้วพบว่าค่าดัชนีชี้วัดมีความใกล้เคียงกันทำให้สามารถสรุปได้ว่าวิธีการทำนายที่เรานำเสนอไปนั้นให้ผลลัพธ์การทำนายและติดตามกำลังผลิตที่ดี

ค่า skill score แสดงให้เห็นว่าผลลัพธ์มีการพัฒนาที่ดีกว่า Persistence Model อยู่ 51.19 เปอร์เซ็นต์ และใกล้เคียงกับกรณีที่ใช้ Artificial Neural Network ในการทำนายที่ได้ค่า skill score อยู่ที่ 53.77 เปอร์เซ็นต์ แต่วิธี ARIMA-Kalman มีข้อดีตรงที่ไม่จำเป็นต้องใช้กำลังผลิตในอดีตหลายวันในการ train แบบจำลองเช่นเดียวกับ ANN

4.3 Case study 2: 5 Solar PV Systems

การทดสอบนี้ใช้จำนวน PV ในระบบ 5 แห่งโดยไม่เกี่ยวข้องกับการทดสอบที่ 4.2 (case study 1: 3 solar PV systems) การทดสอบนี้ใช้ข้อมูลกำลังผลิตตั้งแต่เวลา 5:00 น. ถึงเวลา 18:00 น. โดยที่มีระยะห่างระหว่างข้อมูลเท่ากับ 5 นาที และได้เก็บข้อมูลกำลังผลิตและระยะทางระหว่าง PV แต่ละแห่ง จากนั้นเลือก 1 ใน PV ให้เป็นสถานที่ที่ต้องการทำนายกำลังผลิตเพื่อทดสอบ จากนั้นเลือกวันที่ต้องการทำนายและคำนวณกำลังผลิตแบบเวลาจริงโดยใช้ Estimator model เพื่อใช้ทดสอบกรณีที่ไม่มีการทำนายแบบเวลาจริง กระบวนการทดสอบจะเหมือนกับการทดสอบที่ 4.2 แต่เพิ่มจำนวน PV ในระบบมากขึ้นเพื่อวิเคราะห์ผลลัพธ์

วัตถุประสงค์การทดสอบ ต้องการทดสอบวิธีการทำนายที่เสนอโดยสมมติให้ทำนายกำลังผลิตในวันใดวันหนึ่งภายในชุดข้อมูลที่เก็บมา โดยเปลี่ยนสถานที่และระบบของ PV รวมถึงเพิ่มจำนวน PV ให้มากขึ้นเพื่อศึกษาผลลัพธ์

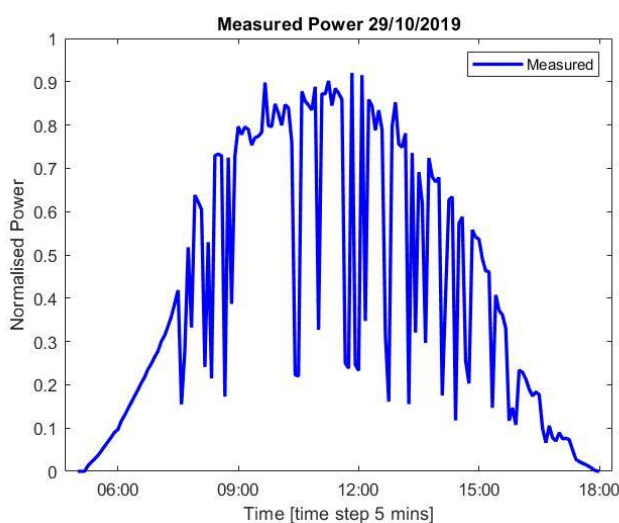
ข้อมูลที่ใช้ในการทดสอบ 1) กำลังผลิตของสถานีทั้ง 5 แห่งโดยกำหนดใช้ชื่อ PV1, PV2, PV3, PV4 และ PV5 โดยที่ข้อมูลกำลังผลิตมีทั้งหมด 1 เดือนและในแต่ละวันมีชุดข้อมูลจำนวน 157 ข้อมูลและข้อมูลมีความละเอียดต่างกันทุก 5 นาที (ในการทดสอบนี้เลือก PV2 เป็น PV ที่ต้องการทำนายกำลังผลิต) ข้อมูลตั้งแต่วันที่ 1/10/2019 ถึง วันที่ 31/10/2019

2) ข้อมูลระยะทางระหว่างสถานที่ที่ต้องการทำนายกำลังผลิต (PV2) ไปยังทุก PV ที่เหลือ

วิธีการทดสอบและผลลัพธ์ของการทดสอบ

4.3.1 Data Inspection

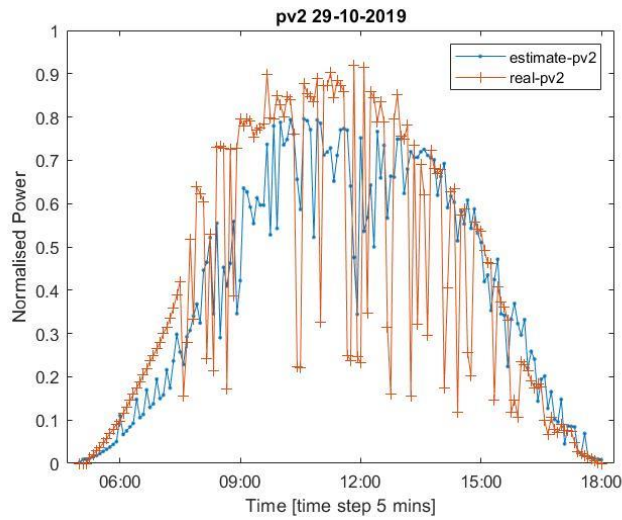
ตรวจสอบ missing data และแก้ไขให้เรียบร้อยจากการประมาณเชิงเส้น จากนั้น plot กราฟกำลังผลิตทั้งหมดที่มีในแต่ละวันเพื่อศึกษาและวิเคราะห์รูปร่างของกราฟและเลือกวันที่จะนำมาเป็นต้นแบบสร้าง ARIMA model โดยกำลังผลิตที่เก็บข้อมูลมีทั้งหมด 1 เดือน ในการทดสอบนี้จะทำนายกำลังผลิตของวันที่ 29/10/2019 รูปที่ 4-24 แสดงกำลังผลิตของวันที่ 29/10/2019 โดยจากการทดสอบที่ 4.1 พบว่าถ้าหากใช้วันที่มีสภาพอากาศแจ่มใสในการสร้างแบบจำลอง ARIMA model จะทำให้ค่าความแม่นยำมีแนวโน้มจะสูงขึ้นเล็กน้อยดังนั้นเราจะเลือกวันที่ 26/10/2019 ซึ่งมีสภาพอากาศแจ่มใสในการสร้างแบบจำลอง ARIMA model



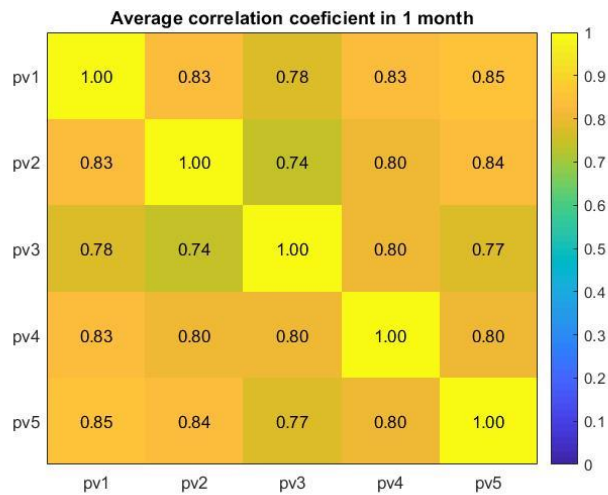
รูปที่ 4-24 รูปภาพกำลังผลิตของวันที่ 29/10/2019

4.3.2 Calculating Power Generation by using Estimator Model

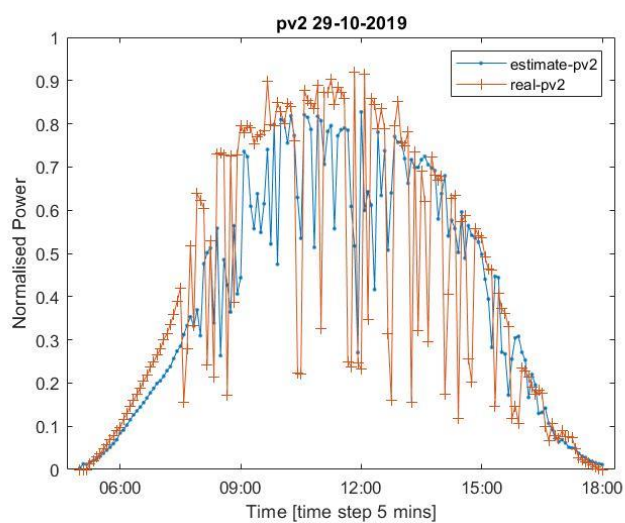
พิจารณารูปภาพกำลังผลิตของแต่ละ PV และคำนวณค่า correlation coefficient ระหว่าง PV เพื่อตรวจสอบว่าค่ากำลังผลิตของแต่ละ PV มีความสัมพันธ์กันมากน้อยเพียงใด ถ้าหากค่ากำลังผลิตของสถานีไหนไม่ค่อยมีความสัมพันธ์กับ PV2 เราจะพิจารณานำ PV นั้นออกจากสมการเพื่อที่ค่าผลลัพธ์จาก estimator model มีคุณภาพที่ดี รูปที่ 4-25 และ 4-26 แสดงกราฟค่าจาก Estimator model กับค่าจริงและแสดงค่าเฉลี่ยของ correlation พบว่ากำลังผลิตของ PV3 ที่เก็บมาได้ไม่สอดคล้องกับ PV2 ในขณะที่ PV1, PV4 และ PV5 มีบางส่วนที่สอดคล้องกับกำลังผลิตของ PV2 ส่งผลให้ค่าเฉลี่ยของ correlation ระหว่าง PV2 กับ PV3 มีค่าน้อยเมื่อเทียบกับค่า correlation ของ PV2 กับ PV อื่น ดังนั้นจึงควรพิจารณานำ PV3 ออกจากสมการการคำนวณ Estimator model หลังจากนั้นจึงคำนวณค่าประมาณกำลังผลิต ในการคำนวณหาค่าประมาณกำลังผลิตแบบเวลาจริงของ PV2 ในวันที่ 29/10/2019 โดยใช้ Estimator model ข้อมูลที่ใช้ในสมการคือข้อมูลระยะทางระหว่าง PV และข้อมูลกำลังผลิตของแต่ละ PV (PV1, PV4 และ PV5 ไม่รวม PV3) ในวันที่ 29/10/2019 ได้ผลลัพธ์ดังรูปที่ 4-27 จะสังเกตได้ว่าเส้นกราฟ estimate-pv2 มีความใกล้เคียงกับเส้นกราฟ real-pv2 มากขึ้น



รูปที่ 4-25 รูปภาพ
เปรียบเทียบค่าประมาณ
จาก Estimator model
กับค่าจริงในวันที่
29/10/2019 ใช้ PV3
คำนวณด้วย



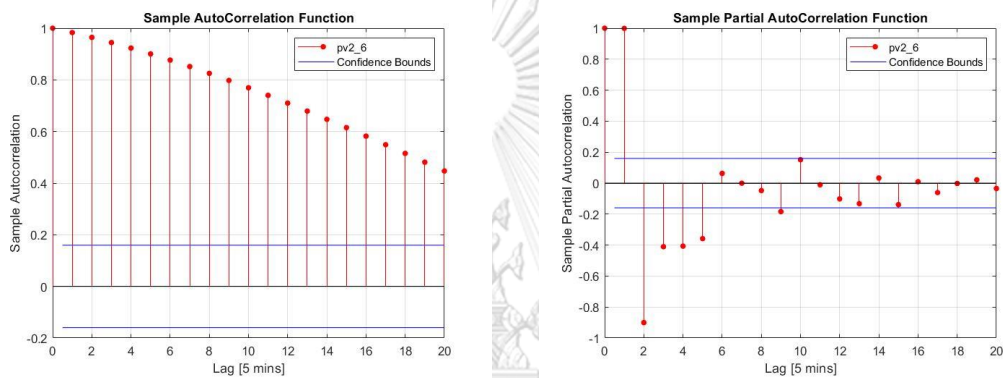
รูปที่ 4-26 ค่าเฉลี่ยของ
correlation ระหว่าง
สถานี (PV1 ถึง PV5) ที่มี
ต่อกันในเวลา 1 เดือน



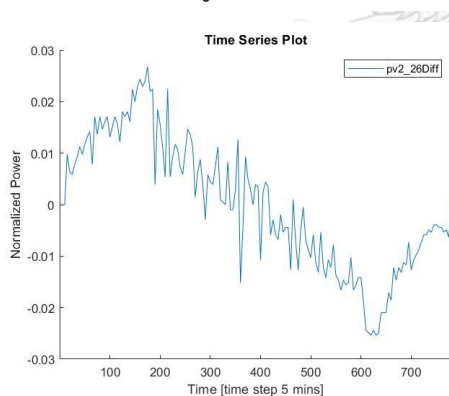
รูปที่ 4-27 รูปภาพ
เปรียบเทียบค่าประมาณ
จาก Estimator model
กับค่าจริงในวันที่
29/10/2019 กรณีที่ไม่
ใช้ PV3 ในการคำนวณ

4.3.3 Created ARIMA Model

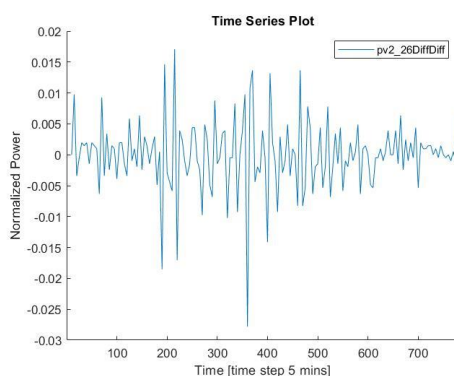
จากข้อมูลกำลังผลิตของ PV2 ในวันที่ 26/10/2019 เรานำข้อมูลมาวิเคราะห์เพื่อสร้าง ARIMA model พบว่าข้อมูลมีความเป็น non-stationary time series จึง plot ACF และ PACF เพื่อตรวจสอบ จากนั้นทำ differencing ข้อมูลได้ผลลัพธ์ดังรูปที่ 4-28 และ 4-29 หลังจากทำ differencing ข้อมูลแล้วพบว่าข้อมูลยังมีความเป็น non-stationary อยู่ดังนั้นจึงทำ differencing อีกครั้งและตรวจสอบ ACF และ PACF ได้ผลลัพธ์ดังรูปที่ 4-30 และ 4-31 พบว่าข้อมูลมีความเป็น stationary มากขึ้นจากนั้นใช้ตารางที่ 2-1 เพื่อระบุค่า p และ q ของแบบจำลองในเบื้องต้น จากนั้นใช้ค่า p, d และ q ที่ทราบสร้างแบบจำลองเพื่อหาค่าสัมประสิทธิ์สำหรับใช้ใน Kalman filter ต่อไป



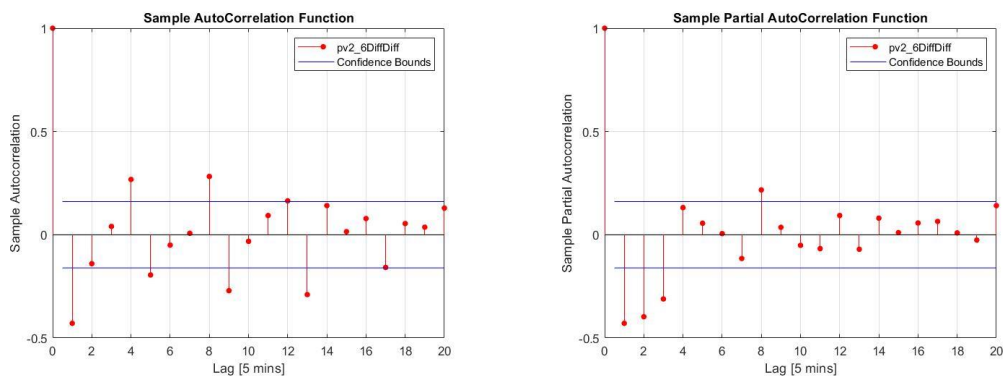
รูปที่ 4-28 ACF plot และ PACF plot ของกำลังผลิต PV2



รูปที่ 4-29 กำลังผลิตหลังจากทำ differencing 1 ครั้ง ของ PV2 ในวันที่ 26/10/2019



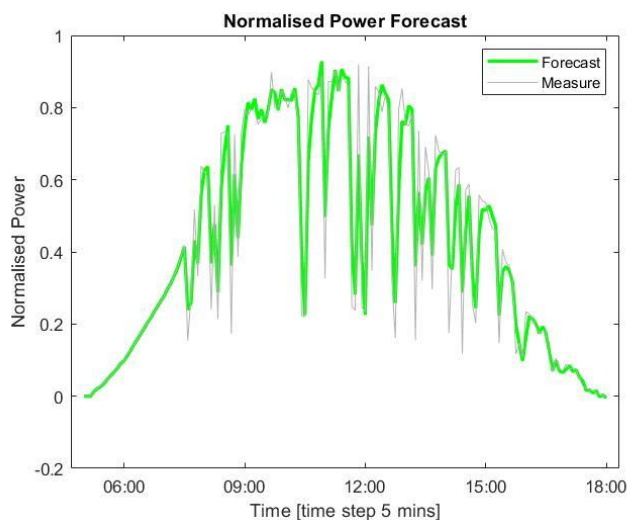
รูปที่ 4-30 กำลังผลิตหลังจากทำ differencing 2 ครั้ง ของ PV2 ในวันที่ 26/10/2019



รูปที่ 4-31 ACF plot และ PACF plot ของกำลังผลิตของ PV2 หลังจากทำ differencing 2 ครั้ง

4.3.4 Forecast without Estimator Model (Observation data is available)

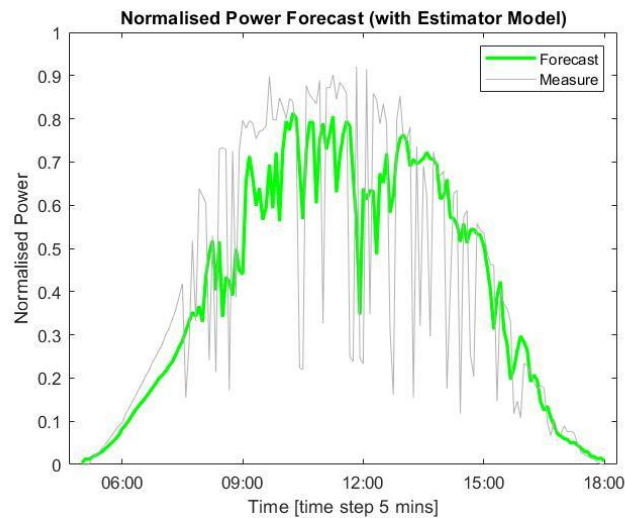
ในกรณีที่สามารถทราบค่ากำลังผลิตแบบเวลาจริงของสถานี PV2 ได้เราจะใช้ค่านั้นใน Kalman Filter algorithm ในขั้นตอนนี้เราจะเขียน ARIMA model ในรูปของ state-space model จากนั้นเราจะใช้ coefficient matrix ของ ARIMA model ในสมการ state forecast ของ Kalman Filter รูปที่ 4-32 แสดงผลลัพธ์ของการทำนายเทียบกับค่าจริง



รูปที่ 4-32 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการทำนายเป็นผลลัพธ์จาก Kalman filter

4.3.5 Forecast with Estimator Model (Observation data is unavailable)

ในกรณีนี้เราใช้ PV ที่เหลืออีก 4 แห่งและระยะทางคำนวณค่าประมาณของกำลังผลิตแบบเวลาจริงของ PV2 จากนั้นจะใช้ค่าประมาณใน Kalman Filter algorithm ซึ่งในการทดสอบถัดไปเราจะมุ่งเน้นที่การคำนวณ Estimator model ที่ใช้ PV ในการคำนวณด้วยระยะทางที่ต่างจากการทดสอบเดิมเพื่อหาข้อสรุปเกี่ยวกับระยะทาง

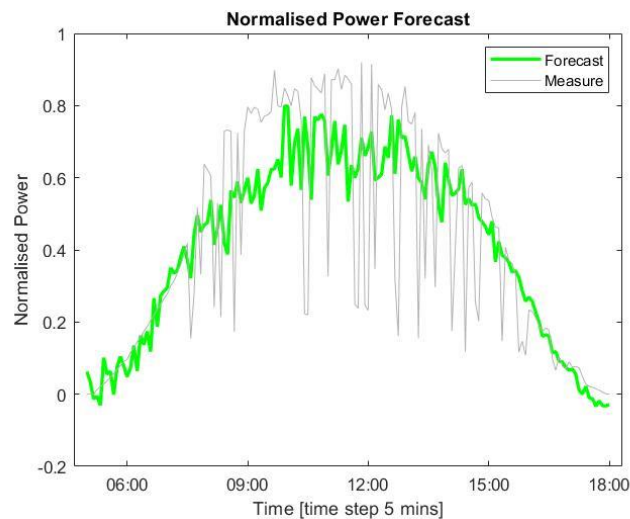


รูปที่ 4-33 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการ
ทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator
model

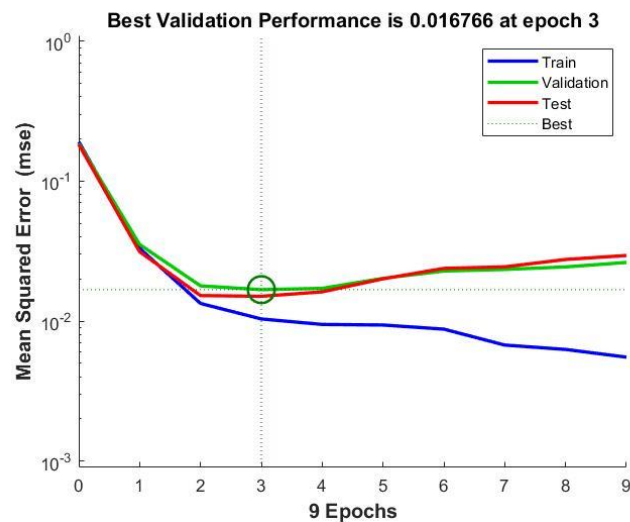
4.3.6 Forecasting by Artificial Neural Network

ในหัวข้อนี้เราจะทำนายกำลังผลิตล่วงหน้า 1 วันโดยใช้วิธีการทำนายที่มีพื้นฐานจาก Artificial Neural Network ตามขั้นตอนที่ได้แสดงให้เห็นในบทที่ 2 หัวข้อ Artificial Neural Network โดยใช้อัลกอริทึมแบบ Levenberg-Marquardt ในการ train แบบจำลองของสถานี PV2 โดยใช้ข้อมูลกำลังผลิตของ PV2 เป็นเวลา 1 เดือนเป็นข้อมูลขาเข้าไปยัง neural network ที่มีจำนวน neuron ใน hidden layer เท่ากับ 10 ในขั้นตอนนี้สามารถทดลอง train แบบจำลองในรูปแบบต่างๆตามที่กล่าวไว้ในบทที่ 2 และเนื่องจากข้อมูลกำลังผลิตที่จำนวนไม่มากสามารถใช้จำนวน hidden layer เท่ากับ 10 เพียงพอต่อการ train แบบจำลอง) ได้ผลลัพธ์ดังรูปที่ 4-34, 4-35 และ 4-36 ผลลัพธ์ระหว่างการทำนายจาก neural network เปรียบเทียบกับค่าจริงจากการวัด แสดงให้เห็นดังรูปที่ 4-34 ในส่วนของค่าจากกราฟ performance หรือค่า MSE ในรูปที่ 4-35 แสดงให้เห็นว่าแบบจำลองที่ train ให้ผลลัพธ์ค่า MSE ที่ดีดูจากเส้นกราฟ Test และ Validation มีความ

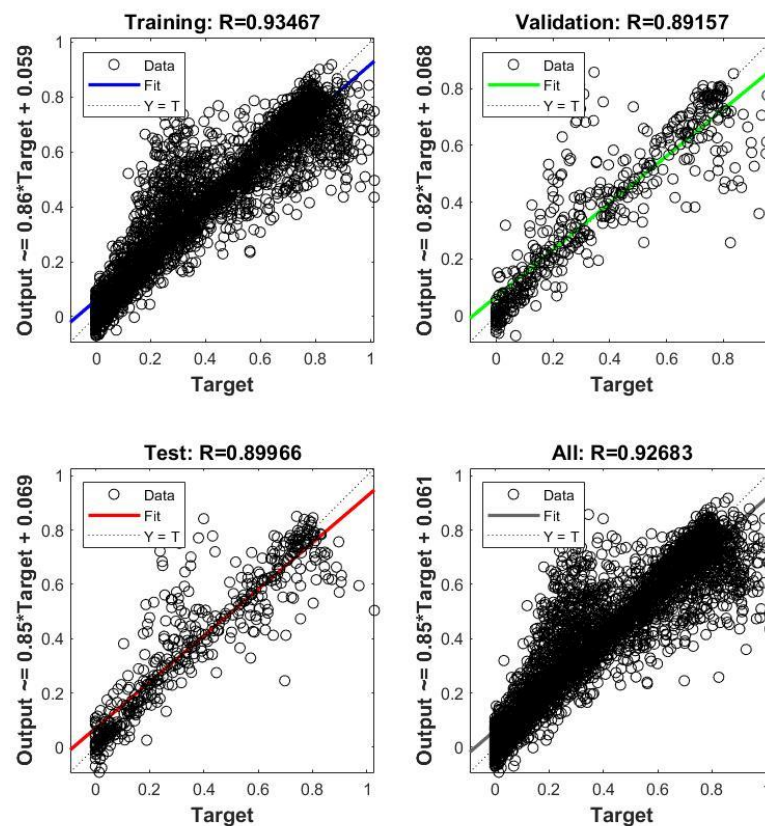
ใกล้เคียงกันมาก นอกจากนั้นกราฟ regression ในรูปที่ 4-36 ยังมีผลลัพธ์ค่า R (ค่า correlation ระหว่างชุดข้อมูลขาออกกับค่าเป้าหมาย) สูงถึง 0.9 (เข้าใกล้ 1) ในกรณีของ Test set และ Validation set แสดงให้เห็นว่าค่าผลลัพธ์ของข้อมูลขาออกเข้าใกล้กับค่าเป้าหมาย



รูปที่ 4-34 กำลังผลิตของสถานี PV2 ในวันที่ 29/10/2019 จากการทำนายโดยใช้ Artificial Neural Network เปรียบเทียบกับค่าจริงที่ได้จากการวัด



รูปที่ 4-35 กราฟค่า performance เปรียบเทียบกันระหว่างชุดข้อมูลที่ใช้ในการ train แบบจำลองของสถานี PV2



รูปที่ 4-36 กราฟ regression เปรียบเทียบกันระหว่างชุดข้อมูลที่ใช้ในการ train

แบบจำลองของสถานี PV2

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

ผลลัพธ์จากในรูปที่ 4-36 เกิดจากการคำนวณค่า correlation ระหว่างผลลัพธ์จากการใช้ชุดข้อมูลต่างๆกับชุดข้อมูลเป้าหมาย โดยที่รูปทางด้านซ้ายบนคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูล training กับชุดข้อมูลเป้าหมาย รูปทางด้านขวาบนคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูล validation กับชุดข้อมูลเป้าหมาย รูปทางด้านซ้ายล่างคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูล test กับชุดข้อมูลเป้าหมาย และ รูปทางด้านขวาล่างคือการ plot ผลลัพธ์จากการใช้ชุดข้อมูลทั้งหมดกับชุดข้อมูลเป้าหมาย โดยที่ชุดข้อมูล training, validation และ test เราได้แบ่งออกมาจากชุดข้อมูลรวมทั้งหมดในขั้นตอน dividing data set ที่ได้กล่าวไปในบทที่ 2 หัวข้อ Artificial Neural Network

4.3.7 Forecasting Results

ในขั้นตอนนี้เราจะวัดค่าความแม่นยำของแต่ละกรณีออกมาเป็นตัวเลขเพื่อความชัดเจน โดยใช้สมการในหัวข้อ การทดสอบค่าความแม่นยำของผลลัพธ์จากแบบจำลองได้แก่ สมการที่ (2.26) ถึง สมการที่ (2.29) ในการคำนวณ โดยเลือกใช้ค่า RMSE, nRMSE และ MSE ผลลัพธ์จากการคำนวณความแม่นยำได้ตามตารางที่ 4-5 (เนื่องจากค่ากำลังผลิตมีค่าเท่ากับ 0 ในบางช่วงที่ไม่มีแสงอาทิตย์ทำให้คำนวณ MAPE ไม่ได้ จึงไม่ได้รวมค่า MAPE ในตารางสรุปและเนื่องจากตัวเลขผลลัพธ์ที่คำนวณได้มีค่าน้อยมากดังนั้นก่อนที่จะคำนวณ RMSE, nRMSE และ MSE เราจะทำให้ค่า normalized กลับเป็นค่าจริงก่อนโดยการคูณด้วยพิกัดกำลังผลิตของ PV2 ซึ่ง PV2 มีพิกัดกำลังผลิตอยู่ที่ 24.6 kW ดังนั้นค่าที่อยู่ในตารางคือผลลัพธ์ที่ได้จากการคูณค่านี้กลับแล้ว)

ตารางที่ 4-5 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิต
ของ PV2 ในวันที่ 29/10/2019

Index	Artificial Neural Network	Kalman Filter with Estimator model	Kalman Filter with real-time data
RMSE	4.4194	4.4747	1.7842
nRMSE	0.4252	0.4305	0.1717
MSE	19.5312	20.0232	3.1834

4.3.8 สรุปผลการทดสอบ

การทดสอบนี้มี PV ทั้งหมด 5 แห่งคือ PV1 ถึง PV5 เราได้เลือก PV2 ในการทดสอบทำนายกำลังผลิต จากนั้นเราใช้ ARIMA model และ Kalman filter ในการทำนายกำลังผลิตแบ่งได้เป็น 2 กรณี กรณีแรกเราใช้กำลังผลิตแบบเวลาจริงของ PV2 ในกระบวนการของ Kalman filter กรณีที่สองเราใช้ Estimator model ในการประมาณค่ากำลังผลิตแบบเวลาจริงเพื่อใช้ใน Kalman filter เมื่อคำนวณผลลัพธ์แล้วพบว่า ความแม่นยำของกรณีที่ใช้กำลังผลิตแบบเวลาจริงมีมากกว่า เห็นได้จากค่า RMSE, nRMSE และ MSE ที่น้อยกว่ากรณีที่สองอย่างเห็นได้ชัดเจน ซึ่งผลลัพธ์เป็นไปตามที่คาดการณ์เนื่องจากเพื่อความต่อเนื่องในระบบการทำนายเราใช้ Kalman filter ในการทำนายและติดตามกำลังผลิตตลอดเวลาดังนั้นผลลัพธ์ในกรณีที่กำลังผลิตแบบเวลาจริงจึงต้องใกล้เคียงกับค่าจริงมากที่สุด ในส่วนของกรณีที่สองเมื่อเราไม่สามารถทราบกำลังผลิตแบบเวลาจริง เมื่อพิจารณาเปรียบเทียบกราฟทำนายกำลังผลิตของกรณีที่สองกับวิธีแบบ Artificial Neural Network แล้วพบว่า

ความแม่นยำใกล้เคียงกัน ซึ่งผลสรุปทั้งสองกรณีนี้สอดคล้องกับการทดสอบก่อนหน้า (การทดสอบที่ 4.2)

ความสำคัญอีกอย่างที่สังเกตได้จากการทดสอบที่ 4.3 คือการที่เพิ่มจำนวน PV ในระบบไม่ได้ก่อให้เกิดผลดีเสมอไป เนื่องจากผลการวิเคราะห์ในการทดสอบนี้เมื่อเรานำ PV3 ออกจากการคำนวณ ทำให้ค่าผลลัพธ์ของ Estimator model (ซึ่งคือค่าประมาณกำลังผลิตแบบเวลาจริง ที่จะนำไปใช้ใน Kalman filter) มีความใกล้เคียงกับความเป็นจริงมากขึ้น สาเหตุเกิดจากกำลังผลิตที่ได้จาก PV3 นั้น เมื่อตรวจสอบโดยการคำนวณค่า correlation แล้วไม่ค่อยสัมพันธ์กับค่ากำลังผลิตของ PV ที่เหลือที่อยู่ในช่วงเวลาเดียวกันนั่นเอง



4.4 การทดสอบผลของระยะทางต่อความแม่นยำของ Estimator Model

วัตถุประสงค์การทดสอบ เพื่อทดสอบว่าระยะทางมีผลต่อการคำนวณโดยใช้ Estimator model อย่างไร และ ระยะทางที่เหมาะสมต่อการคำนวณควรมีค่าประมาณเท่าใด

ข้อมูลที่ใช้ในการทดสอบ

- 1) ข้อมูลกำลังผลิตของ PV1 ถึง PV5 ในการทดสอบที่ 4.3
- 2) ข้อมูลกำลังผลิตที่เก็บเพิ่มโดยตั้งชื่อ PV ที่เก็บข้อมูลเพิ่มว่า PV6 ถึง PV13 (รวมแล้วมี PV ทั้งหมด 13 แห่ง) โดยที่เราเลือกที่จะทำนายกำลังผลิตของ PV2 เหมือนกับแบบทดสอบที่ 4.3
- 3) ข้อมูลระยะทางระหว่าง PV ต่างๆกับ PV2 โดยที่
 - PV1 ถึง PV5 จะมีระยะทางไม่เกิน 10 กิโลเมตรเมื่อเทียบกับ PV2
 - PV6 ถึง PV9 จะมีระยะทางระหว่าง 10 กิโลเมตรถึง 20 กิโลเมตรเมื่อเทียบกับ PV2
 - PV10 ถึง PV13 จะมีระยะทาง 20 กิโลเมตรขึ้นไปเมื่อเทียบกับ PV2

วิธีการทดสอบและผลลัพธ์ของการทดสอบ

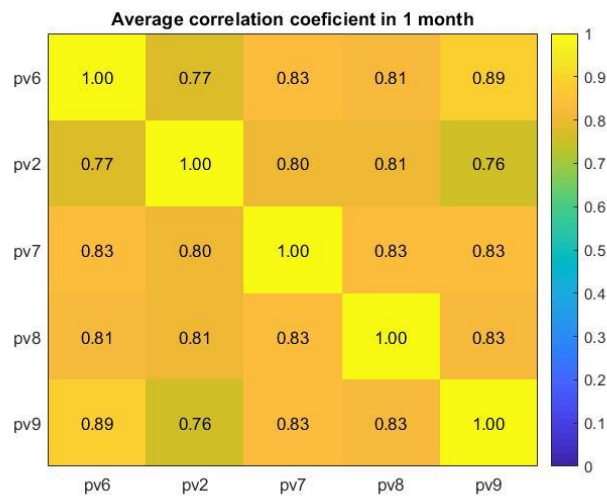
4.4.1 ระยะไม่เกิน 10 กิโลเมตร

การทดสอบระยะไม่เกิน 10 กิโลเมตรจะใช้ผลของการทดสอบที่ 4.3 (ตารางที่ 4-5) เป็นผลลัพธ์เนื่องจากระยะทางของ PV ในการทดสอบที่ 4.3 อยู่ในระยะไม่เกิน 10 กิโลเมตรอยู่แล้ว

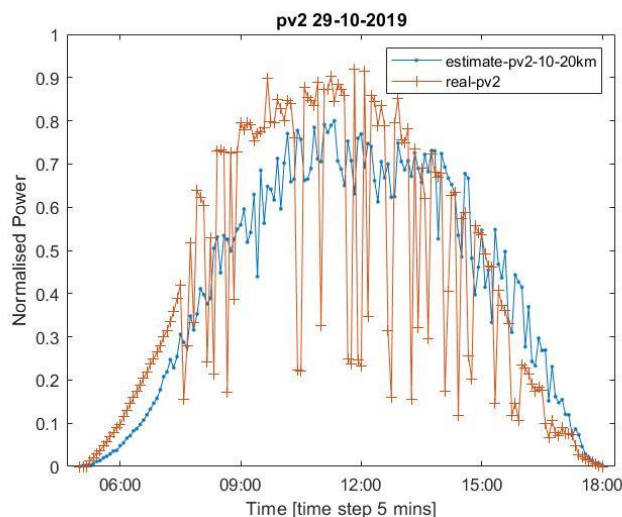
4.4.2 ระยะระหว่าง 10 กิโลเมตรถึง 20 กิโลเมตร

PV ที่ใช้ในการทดสอบได้แก่ PV6, PV7, PV8, PV9 และ PV2 วิธีการทดสอบจะคล้ายกับการทดสอบที่ 4.3 โดยจะใช้ข้อมูลวันที่ 26/10/2019 ในการสร้างแบบจำลอง ARIMA model จากนั้นจะคำนวณหา coefficient ของแบบจำลองเพื่อใช้ใน Kalman filter algorithm ซึ่งจะได้ผลลัพธ์เท่ากับแบบทดสอบก่อนหน้านี้ จากนั้นจะใช้ค่าที่คำนวณได้จาก Estimator model ในกรณีที่ใช้ PV6, PV7, PV8 และ PV9 ในขั้นตอนการคำนวณ Estimator model จากนั้นเมื่อคำนวณค่า correlation ระหว่าง PV ดังรูปที่ 4-37 พบว่าผลลัพธ์ของค่า correlation ระหว่างคู่ของ PV2 กับ PV6 และ PV2 กับ PV9 มีค่าน้อยกว่าค่า correlation ระหว่าง PV อื่นในขอบเขตระยะ 10 กิโลเมตรถึง 20 กิโลเมตร ดังนั้นจึงพิจารณานำ PV6 และ PV9 ออกไปจากสมการ Estimator model และใช้เพียง PV7 และ PV8 ในการคำนวณ

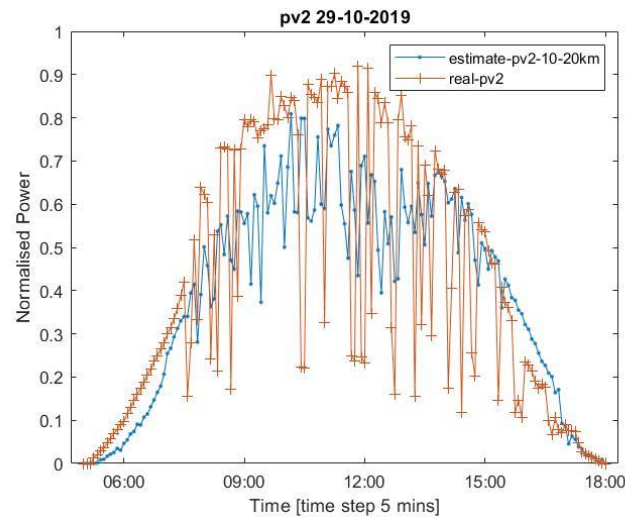
รูปที่ 4-38 แสดงถึงการเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงและรูปที่ 4-39 แสดงการเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงเมื่อเรานำ PV6 และ PV9 ออกจากการคำนวณของสมการ Estimator model ต่อจากนั้นจึงป้อนข้อมูลที่คำนวณได้เข้าใน Kalman filter เพื่อทำนายกำลังผลิตได้ผลลัพธ์การทำนายดังรูปที่ 4-40 และเมื่อคำนวณดัชนีชี้วัดได้ผลลัพธ์ดังตารางที่ 4-6



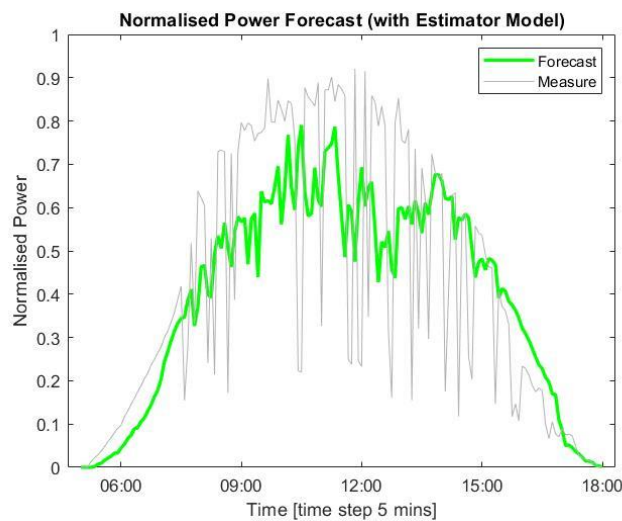
รูปที่ 4-37 ค่าเฉลี่ยของ correlation ของสถานี (PV6 ถึง PV9 และ PV2) ที่มีต่อกันในเวลา 1 เดือน



รูปที่ 4-38 รูปภาพเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ใช้ PV6 และ PV9 ร่วมในการคำนวณ



รูปที่ 4-39 รูปภาพเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ไม่ใช่ PV6 และ PV9 ร่วมในการคำนวณ



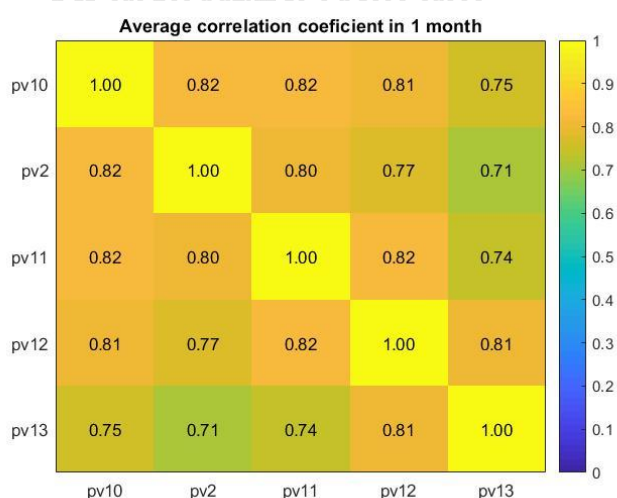
รูปที่ 4-40 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator model คำนวณค่าประมาณข้อมูลกำลังผลิตแบบเวลาจริง กรณีระยะทางระหว่าง 10-20 กิโลเมตร

ตารางที่ 4-6 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิต
ของ PV2 ในวันที่ 29/10/2019 ในระยะทางระหว่าง 10-20 กิโลเมตร

Index	Kalman Filter with Estimator model
RMSE	4.8313
nRMSE	0.4649
MSE	23.3418

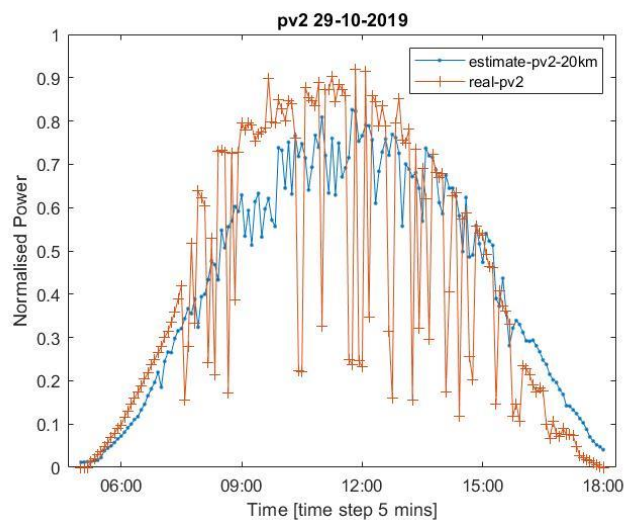
4.4.3 ระยะทางตั้งแต่ 20 กิโลเมตรขึ้นไป

PV ที่ใช้ในการทดสอบได้แก่ PV10, PV11, PV12, PV13 และ PV2 การทดสอบคล้ายกับการทดสอบที่ 4.3 โดยจะใช้ข้อมูลวันที่ 26/10/2019 ในการสร้างแบบจำลอง ARIMA model จากนั้นจะคำนวณหา coefficient ของแบบจำลองเพื่อใช้ใน Kalman filter algorithm ซึ่งจะได้ผลลัพธ์เดียวกับการทดสอบก่อนหน้านี้ จากนั้นจะใช้ค่าที่คำนวณได้จาก Estimator model ในกรณีที่ใช้ PV10, PV11, PV12 และ PV13 ในขั้นตอนการคำนวณ Estimator model พบว่าเมื่อวาดกราฟและคำนวณค่า correlation ระหว่าง PV ดังรูปที่ 4-41 พบว่าผลลัพธ์ของค่า correlation ระหว่าง PV2 กับ PV13 มีค่าน้อยกว่าค่า correlation ระหว่าง PV อื่นพอสมควร นอกจากนั้นกราฟกำลังผลิตในแต่ละวันไม่สอดคล้องกับอื่นมาก ดังนั้นจึงพิจารณานำ PV13 ออกไปจากสมการ Estimator model และใช้เพียง PV10, PV11 และ PV12 ในการคำนวณ

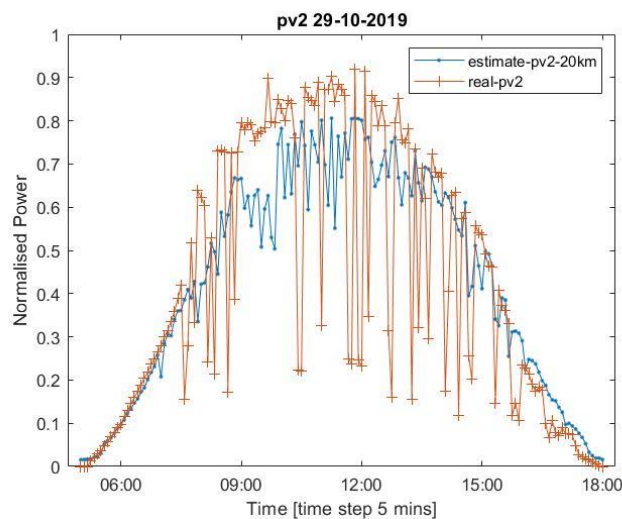


รูปที่ 4-41 ค่าเฉลี่ยของ correlation ของสถานี (PV10 ถึง PV13 และ PV2) ที่มีต่อกันในเวลา 1 เดือน

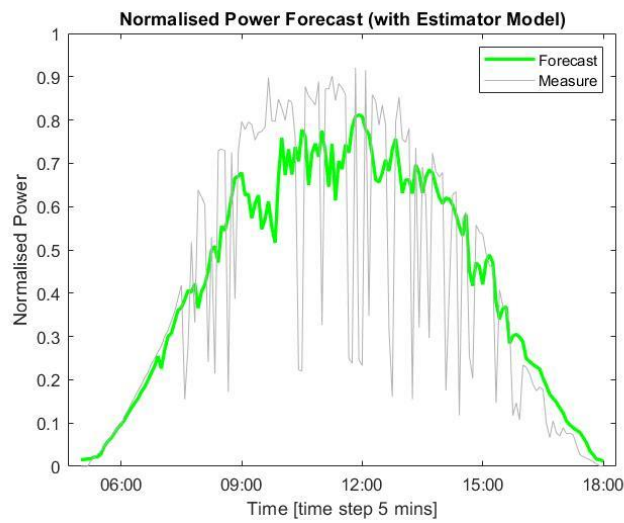
รูปที่ 4-42 แสดงถึงการเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงและรูปที่ 4-43 แสดงการเปรียบเทียบค่าประมาณจาก Estimator model กับค่าจริงเมื่อเรานำ PV13 ออกจากการคำนวณของสมการ Estimator model ต่อจากนั้นจึงป้อนข้อมูลที่คำนวณได้เข้าใน Kalman filter เพื่อทำนายกำลังผลิตได้ผลลัพธ์การทำนายดังรูปที่ 4-44 และเมื่อคำนวณดัชนีชี้วัดได้ผลลัพธ์ดังตารางที่ 4-7



รูปที่ 4-42 รูปภาพเปรียบเทียบค่าจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ใช้ PV13 ในการคำนวณ



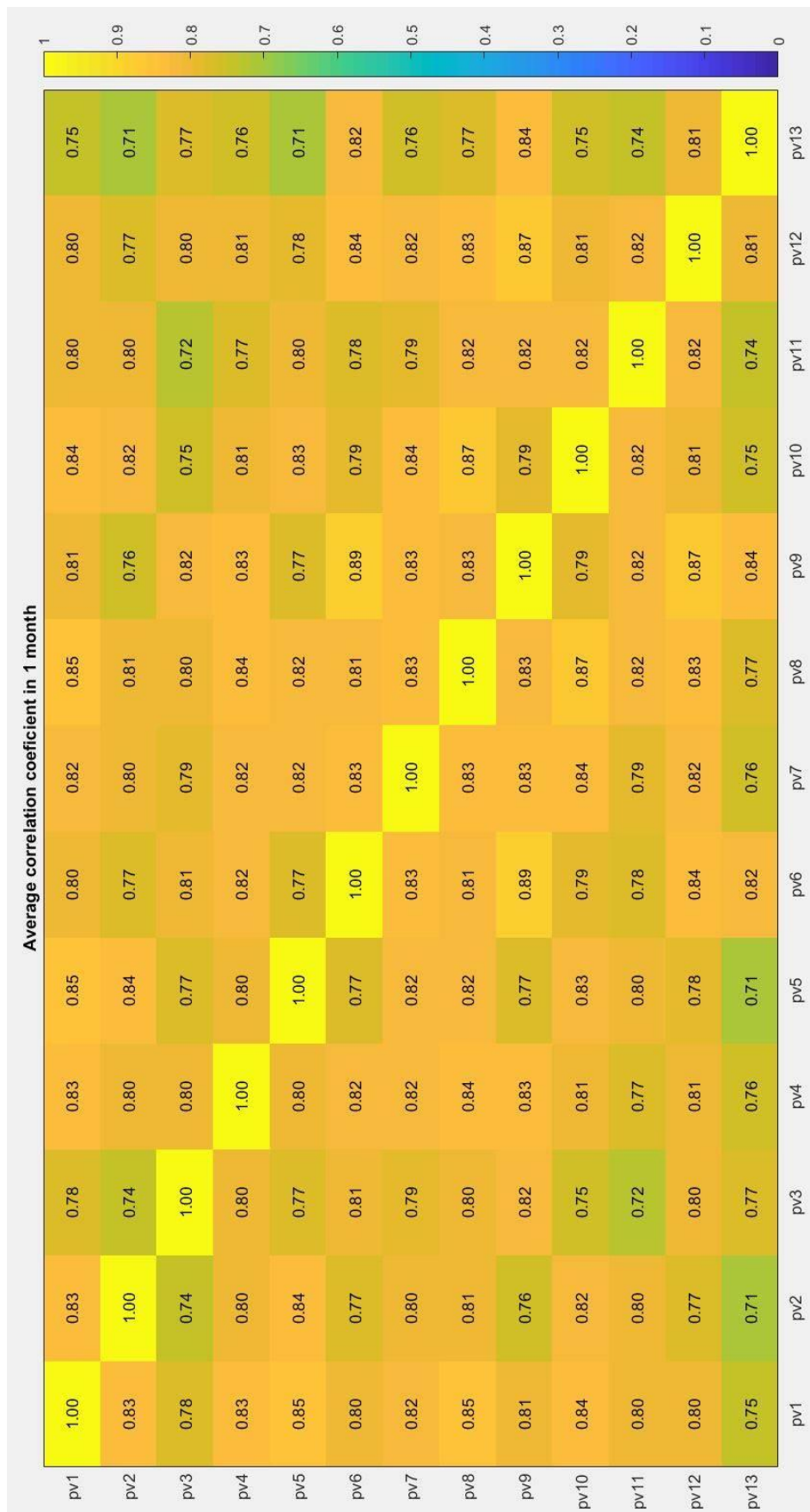
รูปที่ 4-43 รูปภาพเปรียบเทียบค่าจาก Estimator model กับค่าจริงในวันที่ 29/10/2019 ไม่ใช้ PV13 คำนวณ



รูปที่ 4-44 กำลังผลิตของ PV2 ในวันที่ 29/10/2019 จากการ
ทำนายเป็นผลลัพธ์จาก Kalman filter โดยใช้ Estimator
model คำนวณค่าประมาณข้อมูลกำลังผลิตแบบเวลาจริง กรณี
ระยะทางเกิน 20 กิโลเมตร

ตารางที่ 4-7 ตารางชี้วัดความแม่นยำของผลลัพธ์จากการทำนายกำลังผลิตของ PV2 ในวันที่
29/10/2019 ในระยะทางมากกว่า 20 กิโลเมตร

Index	Kalman Filter with Estimator model
RMSE	4.6929
nRMSE	0.4515
MSE	22.0237



รูปที่ 4-45 ค่า correlation ระหว่าง PV ทั้งหมดในระบบ

4.4.4 สรุปผลการทดสอบ

จากการทดสอบด้านระยะทางพบว่าเมื่อพิจารณาค่า correlation ของแต่ละ PV ที่ทดสอบ ระยะทางในรูปที่ 4-45 จะเห็นได้ว่ามีแนวโน้มที่ลดลงเมื่อระยะห่างระหว่าง PV กับ PV2 มากขึ้น สังเกตได้จาก PV1, PV2, PV3 และ PV5 ภายในชุดนี้จะมีเพียง PV3 เท่านั้นที่มีค่า correlation กับ PV2 น้อย และเมื่อพิจารณาชุดถัดไป PV6, PV7, PV8 และ PV9 ภายในชุดนี้มี PV ที่มีค่า correlation กับ PV2 น้อย ถึง 2 PV นั่นก็คือ PV6 และ PV9 เช่นเดียวกับชุดข้อมูลของกลุ่มที่มี ระยะทางมากกว่า 20 กิโลเมตรขึ้นไป แสดงให้เห็นว่าเมื่อระยะทางเพิ่มขึ้นจะมีโอกาสที่เราจะพบ PV ที่มีความไม่สัมพันธ์กับ PV ที่เราต้องการทำนายกำลังผลิตมากขึ้นจึงสามารถสรุปได้ว่า PV ที่อยู่ใกล้ PV2 มากควรนำมาคำนวณใน Estimator model มากกว่า PV ที่อยู่ไกล โดยเมื่อพิจารณาจากกรณีที่เราได้ทดสอบจึงสรุปได้ว่า PV ที่อยู่ใกล้ PV2 ไม่เกิน 10 กิโลเมตรสามารถใช้ในการคำนวณได้ดีกว่า ในกรณีที่อยู่ไกลกว่า 10 กิโลเมตร นอกจากนั้นการทดสอบค่า correlation มีความจำเป็นต้องทำ ก่อนการคำนวณ Estimator model เนื่องจากค่าประมาณกำลังผลิตจาก Estimator model ส่งผล ต่อผลลัพธ์ของการทำนายอย่างมาก

ในส่วนของตารางดัชนีชี้วัดความแม่นยำของตารางที่ 4-6 และ ตารางที่ 4-7 พบว่ามีค่า RMSE, nRMSE และ MSE ที่มากกว่าตารางที่ 4-5 แสดงให้เห็นว่าเราทำนายกำลังผลิตของ PV ที่มี ระยะทางไม่เกิน 10 กิโลเมตรได้ผลลัพธ์ดีกว่าสอดคล้องกับข้อสรุปทางด้าน correlation ข้างต้น

บทที่ 5

สรุปผลการทดสอบ

5.1 บทสรุป

วิทยานิพนธ์ฉบับนี้นำเสนอแนวทางในการทำนายกำลังผลิตของ PV โดยสามารถแบ่งกระบวนการคำนวณได้เป็น 3 ส่วนใหญ่คือ 1. การสร้างแบบจำลองการทำนายโดยใช้ ARIMA Model 2. การทำนายหรือติดตามกำลังผลิตของ PV แบบเวลาจริงโดยใช้ Kalman filter algorithm 3. การคำนวณค่าประมาณกำลังผลิตแบบเวลาจริงเพื่อใช้ใน Kalman filter algorithm โดยใช้ Estimator model เริ่มต้นที่ส่วนแรกการสร้างแบบจำลอง ARIMA model เราได้เก็บข้อมูลกำลังผลิตในอดีตของ PV ที่เราต้องการทำนายและ PV ช่างเคียงเพื่อนำมาวิเคราะห์รูปแบบของกำลังผลิตและพิจารณาพฤติกรรมของกำลังผลิต จากนั้นเราใช้ข้อมูลกำลังผลิตของ PV ที่ต้องการทำนายกับกระบวนการของ Box & Jenkins เพื่อระบุค่า parameter ที่จำเป็นในแบบจำลอง ARIMA model จากนั้นเราประมาณค่าสัมประสิทธิ์ของตัวแปรในแบบจำลองแต่ละตัวเพื่อนำไปใช้ใน Kalman filter algorithm ในส่วนที่สองการทำนายโดยใช้ Kalman filter ในส่วนนี้เราจำเป็นต้องใช้ข้อมูลกำลังผลิตแบบเวลาจริงของ PV ที่ต้องการทำนาย ซึ่งเราได้แนะนำเสนอให้ใช้ Estimator model ในการคำนวณค่าประมาณกำลังผลิตแบบเวลาจริงในกรณีที่เราไม่สามารถทราบค่ากำลังผลิตจาก PV ที่ต้องการทำนายได้ ในส่วนที่สามคือกระบวนการคำนวณค่าประมาณกำลังผลิตแบบเวลาจริงโดยใช้ Estimator model แบบจำลองที่เราได้นำเสนอนี้จะใช้ข้อมูลกำลังผลิตแบบเวลาจริงของ PV ช่างเคียงร่วมกับข้อมูลของระยะทาง เพื่อประมาณค่ากำลังผลิตของ PV ที่เราสนใจ เรานำค่าผลลัพธ์จาก Estimator model ไปใช้ใน Kalman filter algorithm ได้โดยที่ไม่ทำให้การทำนายขาดช่วง

นอกจากนี้ในวิทยานิพนธ์ฉบับนี้ได้ทดสอบกระบวนการที่ได้นำเสนอมาในบทที่ 4 โดยได้แบ่งการทดสอบเป็น 4 การทดสอบคือการทดสอบที่ 4.1, 4.2, 4.3 และ 4.4 ในการทดสอบที่ 4.1 เราได้ทดสอบศึกษาผลลัพธ์ของการเลือกใช้ชุดข้อมูลในการ fit ARIMA model และสภาพอากาศของวันที่ต้องการทดสอบมีผลอย่างไรต่อความแม่นยำในการทำนายกำลังผลิต โดยได้แบ่งวันตามสภาพอากาศออกเป็น 3 แบบ 1. วันที่มีอากาศแจ่มใส 2. วันที่มีสภาพอากาศมีเมฆมาก 3. วันที่มีสภาพอากาศฝนตก และได้นำวันทั้ง 3 นี้มา fit ARIMA model แล้วทำนายกำลังผลิตพบว่าวันทั้ง 3 รูปแบบให้ผลลัพธ์ที่มีความแม่นยำใกล้เคียงกัน จากนั้นเพื่อยืนยันผลการทดสอบเราได้นำวันทั้ง 3 แบบนี้ไปทำนายกำลังผลิตของวันต่างๆที่เราเก็บข้อมูลมาได้เป็นเวลา 1 เดือน พบว่าเมื่อเราวาดกราฟ box plot แล้วถ้าเรานำวันทั้ง 3 ไปทำนายกับวันที่มีสภาพอากาศฝนตกจะให้ผลลัพธ์ที่มีความคลาดเคลื่อนมากกว่านำไปทำนายกับวันที่มีสภาพอากาศมีเมฆมาก อีกทั้งวันทั้ง 3 รูปแบบนั้นยังให้ผลลัพธ์ค่าเฉลี่ย

และช่วงของควาคลาดเคลื่อนที่ใกล้เคียงกันสอดคล้องกับผลการทดสอบเบื้องต้นตั้งนั้นสามารถสรุปได้ว่าชนิดวันที่นำมา fit ARIMA model มีผลกับผลลัพธ์ของการทำนายไม่ต่างกันมากนัก ผลลัพธ์ของการทำนายจะขึ้นกับสภาพอากาศวันที่เราต้องการที่จะทำนายกำลังผลิตและผลลัพธ์การประมาณค่ากำลังผลิตแบบเวลาจริงของ Estimator model มากกว่าซึ่งแสดงว่า Estimator model สามารถประมาณค่ากำลังผลิตแบบเวลาจริงจากการใช้ข้อมูลกำลังผลิตของ PV รอบข้างได้ค่อนข้างแม่นยำ

ในการทดสอบที่ 4.2 เราได้แสดงขั้นตอนวิธีการทำนายที่เราเสนอโดยใช้ข้อมูลรายละเอียดของ PV จากการทดลองที่ 4.1 และนำไปเปรียบเทียบกับ Persistence Model และ Artificial Neural Network ผลการทดสอบพบว่ากรณีที่ใช้ Estimator model คำนวณกำลังผลิตแบบเวลาจริงก่อนแล้วจึงใช้ค่านั้นร่วมกับ Kalman Filter สามารถใช้ประมาณค่ากำลังผลิตแบบเวลาจริงและนำมาใช้กับ Kalman Filter ได้โดยที่ผลลัพธ์อยู่ในระดับที่ดีและแสดงให้เห็นถึงการเคลื่อนที่ของเส้นกราฟตามเส้นกราฟกำลังผลิตที่แท้จริง นอกจากนี้เมื่อเปรียบเทียบกับการทำนายแบบที่ใช้ Artificial Neural Network แล้วพบว่าค่าดัชนีชี้วัดมีความใกล้เคียงกัน อีกทั้งผลลัพธ์จากทั้ง 2 วิธียังดีกว่า Persistence Model อีกด้วย ซึ่งตามปกติแล้ว Persistence Model เมื่อใช้กับการทำนายระยะสั้นแล้วจะให้ผลลัพธ์ที่ดีใกล้เคียงกับการทำนายที่ใช้วิธีที่ซับซ้อนมากเนื่องจากลักษณะของสภาพอากาศที่บางครั้งไม่เปลี่ยนแปลงทำให้ Persistence Model มีความสามารถที่สูงหากทำนายในวันที่สภาพอากาศค่อนข้างคงที่ ทำให้สามารถสรุปได้ว่าวิธีการทำนายที่เราเสนอไปนั้นให้ผลลัพธ์การทำนายและติดตามกำลังผลิตที่ดี นอกจากนี้วิธีการ ARIMA-Kalman ยังไม่จำเป็นต้องใช้ข้อมูลของสถานี PV ที่เราสนใจทำให้ลดภาระในการเก็บข้อมูลได้

ในการทดสอบที่ 4.3 เราได้เปลี่ยนระบบของ PV เติมโดยการเพิ่มจำนวน PV ในระบบให้มากขึ้นและใช้หลักการเติมในการทดสอบ พบว่าการเพิ่มจำนวน PV ในการคำนวณ Estimator model อาจส่งผลเสียในกรณีที่ PV นั้นมีคุณภาพของกำลังผลิตไม่ดีทำให้รูปร่างของกราฟกำลังผลิตไม่ค่อยแม่นยำ ดังนั้นเราจึงควรตรวจสอบความสัมพันธ์ของค่ากำลังผลิตในระบบ PV ก่อนแล้วจึงคำนวณ Estimator model เพราะค่าประมาณกำลังผลิตจาก Estimator model มีความสำคัญต่อผลลัพธ์ในการทำนายค่อนข้างมาก อีกทั้งการเพิ่มจำนวน PV มากเกินไปจะทำให้การคำนวณ Estimator model ไม่แม่นยำและใช้เวลานานมากขึ้น ดังนั้นจึงควรใช้ PV ไม่เกิน 5 แห่งในระบบการคำนวณ

การทดสอบที่ 4.4 ทดสอบเกี่ยวกับระยะทางของ PV ที่ใช้ในการคำนวณของ Estimator model ว่ามีผลกับการทำนายอย่างไร โดยผลลัพธ์ได้แบ่งเป็น 3 ระยะ 1. ไม่เกิน 10 กิโลเมตร 2. ระหว่าง 10-20 กิโลเมตร 3. เกิน 20 กิโลเมตร พบว่าค่า correlation ของแต่ละ PV ที่ทดสอบมีแนวโน้มที่ลดลงเมื่อระยะห่างระหว่าง PV นั้นกับ PV2 มากขึ้น แสดงให้เห็นว่าระยะทางมีผลต่อการคำนวณ เมื่อพิจารณาจากกรณีที่เราได้ทดสอบจึงสรุปได้ว่าระยะที่เหมาะสมไม่ควรเกิน 10 กิโลเมตร หรือถ้าใช้ PV ที่ระยะเกิน 10 กิโลเมตรควรตรวจสอบค่าความสัมพันธ์โดยใช้ correlation ให้แน่ใจ และตรวจสอบกราฟกำลังผลิตให้แน่ใจว่าในเวลาช่วงเดียวกันกราฟมีลักษณะคล้ายกันเป็นส่วนใหญ่ เนื่องจากระยะทางที่มากเกินไปส่งผลให้สภาพอากาศของแต่ละ PV ไม่เหมือนกัน

5.2 ข้อเสนอแนะ

1. ควรตรวจสอบคุณภาพของ PV ที่เก็บค่ากำลังผลิตมาว่ามีสิ่งผิดปกติหรือไม่และมีข้อมูลกำลังผลิตที่หายไปหรือไม่
2. ไม่ควรใช้ PV จำนวนมากในการคำนวณ Estimator model เนื่องจากผลของการรวมกันของค่ากำลังผลิตของ PV แต่ละแห่งไม่ได้ทำให้ค่ากำลังผลิตที่ประมาณจาก Estimator model ดีขึ้นเสมอไปดังที่ได้นำเสนอไปในการทดสอบที่ 4.3 ควรเลือก PV ไม่เกิน 5 แห่งและเมื่อตรวจสอบค่า correlation แล้วมีค่ามากกว่าหรือเท่ากับ 0.8 (ทั้งการทดสอบกับ PV ที่ต้องการทำนายกำลังผลิตและ PV ข้างเคียงอื่นๆในระบบ) อย่างไรก็ตามผลลัพธ์ไม่แน่นอนเสมอไปว่าควรใช้ PV ก็เพิ่มขึ้นอยู่กับคุณภาพของข้อมูลกำลังผลิตที่วัดได้ของ PV นั้นๆด้วย
3. ไม่ควรใช้ PV ที่มีระยะทางไกลจาก PV ที่ต้องการทำนายกำลังผลิตเกิน 10 กิโลเมตร เนื่องจากในบางครั้งสภาพอากาศอาจทำให้ค่ากำลังผลิตที่วัดได้บังเอิญเหมือนกันซึ่งมีโอกาสเกิดขึ้นได้น้อยกว่ากรณีที่ PV อยู่ภายในระยะ 10 กิโลเมตร

บรรณานุกรม

- [1] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. Martinez-de-Pison, and F. J. S. E. Antonanzas-Torres, "Review of photovoltaic power forecasting," *Solar Energy*, vol. 136, pp. 78-111, 2016.
- [2] C. Voyant *et al.*, "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569-582, 2017.
- [3] S. Pelland, G. Galanis, G. J. P. i. P. R. Kallos, and Applications, "Solar and photovoltaic forecasting through post - processing of the Global Environmental Multiscale numerical weather prediction model," *Progress in Photovoltaics: Research and Applications*, vol. 21, no. 3, pp. 284-296, 2013.
- [4] M. Diagne, M. David, J. Boland, N. Schmutz, and P. J. S. E. Lauret, "Post-processing of solar irradiance forecasts from WRF model at Reunion Island," *Solar Energy*, vol. 105, pp. 99-108, 2014.
- [5] T. Soubdhan, J. Ndong, H. Ould-Baba, and M.-T. J. S. E. Do, "A robust forecasting framework based on the Kalman filtering approach with a twofold parameter tuning procedure: Application to solar and photovoltaic prediction," *Solar Energy*, vol. 131, pp. 246-259, 2016.
- [6] M. Hassanzadeh, M. Etezadi-Amoli, and M. Fadali, "Practical approach for sub-hourly and hourly prediction of PV power output," in *North American Power Symposium (NAPS) 2010*, 2010, pp. 1-5: IEEE.
- [7] L. A. Fernandez-Jimenez *et al.*, "Short-term power forecasting system for photovoltaic plants," *Renewable Energy*, vol. 44, pp. 311-317, 2012/08/01/ 2012.
- [8] A. Mellit, A. Massi Pavan, and V. Lughi, "Short-term forecasting of power production in a large-scale photovoltaic plant," *Solar Energy*, vol. 105, pp. 401-413, 2014/07/01/ 2014.
- [9] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	พีรพล จิรนนทเจริญ
วัน เดือน ปี เกิด	21 กุมภาพันธ์ 2538
สถานที่เกิด	กรุงเทพฯ
ที่อยู่ปัจจุบัน	23 ซ.เจริญรัถ 4 ถ.เจริญรัถ แขวงคลองตันใต้ เขตคลองสาน กรุงเทพฯ 10600



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY