

공학석사 학위논문

태양광 패널의 데이터시트 기반 모델링
및 응용

Datasheet-based Modeling of Photovoltaic Panel
and Its Applications

울산대학교 대학원
전기공학부
박준영

태양광 패널의 데이터시트 기반 모델링 및 응용

지도교수 최성진

이 논문을 공학석사 학위논문으로 제출함

2015 년 12 월

울산대학교 대학원

전기공학부

박준영

박준영의 공학석사학위 논문을 인준함

심사위원장 이 홍 희 인

심 사 위 원 전 태 원 인

심 사 위 원 최 성 진 인

울 산 대 학 교 대 학 원

2015 년 12 월

[국문요약]

태양광 패널의 데이터시트 기반 모델링 및 응용

울산대학교 대학원
전 기 공 학 부
박 준 영

태양광 시스템의 개발과정에서 시스템에 대한 검증이 필요하지만 시스템이 구동되는 주 변의 온도나 매 순간의 일사량에 따라서 실제 태양광 패널의 출력이 실시간으로 변하기 때문에 개발과정에서 실제 패널을 사용하는 것이 어렵다. 그래서 시스템의 검증과정 중에 서 소프트웨어를 통한 시뮬레이션 단계와 시뮬레이터를 이용한 하드웨어과정이 필요하며, 따라서 태양광 패널의 출력특성을 잘 모사할 수 있는 전기적 등가회로에 대한 연구가 활 발히 진행되고 있다.

본 논문은 태양광 패널의 특성을 잘 표현하는 전기적 등가모델을 제안하며, 제안모델은 PSIM을 통해서 소프트웨어적으로 사용이 가능하며, 시스템의 엔진으로서 하드웨어 실험에 도 적용이 될 수 있다. 이를 위해서 본 논문은 기존의 등가모델을 구하는 모델링 방법들을 조사하였고 각 방법의 장단점을 조사하여 이를 보완하는 방향으로 제안방법을 제시하였다. 그 결과 제안방법은 데이터시트만을 이용해서 등가회로 모델링을 하며, 이때 가변환경에서 의 모델의 정확도를 향상시키기 위해서 동적 데이터시트를 도입하였다. 제안모델의 타당성 의 검증은 EN50530의 유럽규격을 따라서 진행하였고 그 결과를 기존의 방법과 비교하였 다. 소프트웨어구현은 PSIM을 통해서 통합회로를 구성하여 사용자의 사용성과 편리성을 증대시켰으며, 하드웨어구현은 DC서플라이와 벽-컨버터를 이용하는 회로로 출력특성이 실 제 패널과 동일하게 모사되는 것을 시뮬레이션으로 검증하였다. 최종적으로 본 논문은 소 프트웨어 및 하드웨어 모두에서 사용이 가능한 새로운 태양광 등가회로 모델을 제안한다.

목 차

국문요약	I
목 차	II
그림목차	IV
표 목 차	VI
1. 서 론	1
1.1. 연구배경 및 필요성	1
1.2. 태양광 패널의 특성과 등가모델	2
1.3. 연구목표 및 내용	4
2. 기존의 데이터시트 기반의 모델링 방법	6
2.1. 데이터시트 기반의 모델링	6
2.1.1. 데이터시트 기반 모델링의 장점과 주요이슈	6
2.1.2. 데이터시트의 정보부족 문제	6
2.2. 기존의 모델링 방법 분석	9
2.2.1. 파라미터 제거법	9
2.2.2. 파라미터 고정법	10
2.2.3. 기울기 조건 추가법	11
3. 제안하는 데이터시트 기반의 방법	14
3.1. 제안방법의 수식전개	15
3.1.1. 목표함수 설정	15
3.1.2. I_{ph} 및 I_o 수식유도	17
3.2. Powell 최적화 방법	18
3.3. 탐색범위 및 초기값 설정	22
4. 제안방법의 검증 및 응용	24
4.1. 제안방법의 검증	24
4.1.1. MATLAB과 PSIM을 이용한 모델링	24

4.1.2.	EN50530 규격	27
4.1.3.	결과비교	33
4.2.	제안방법의 소프트웨어 응용	35
4.2.1.	제안모델의 가변환경 모델링	35
4.2.2.	PSIM의 태양광 모델	36
4.2.3.	C block을 이용한 제안모델의 구현	44
4.2.4.	PSIM 모델과의 비교검증	46
4.3.	제안방법의 하드웨어 응용	51
4.3.1.	벽-컨버터 전력단 설계	52
4.3.2.	벽-컨버터의 소신호 모델링	53
4.3.3.	제어기 설계전략	55
4.3.4.	전류 제어기의 설계	60
4.3.5.	제어기 시뮬레이션 검증	62
4.3.6.	PV 특성모사 성능평가	65
5.	결 론	69
	참 고 문 헌	70
	부 록	75
	ABSTRACT	94
	연구논문 실적	95

그림 목 차

그림 1.1	싱글 다이오드 등가모델	2
그림 1.2	태양광 패널의 출력특성	3
그림 2.1	태양광 패널의 특성지점	7
그림 2.2	데이터시트의 정보부족 문제	8
그림 2.3	파라미터 제거방법	9
그림 2.4	파라미터 고정방법	10
그림 2.5	파라미터 고정방법의 단점	11
그림 2.6	파라미터 추출을 위한 추가조건	12
그림 2.7	조건 추가법의 단점	12
그림 3.1	제안방법의 개념도	14
그림 3.2	파wel의 최적화 방법의 수도코드	19
그림 3.3	파wel의 최적화 방법의 동작원리	20
그림 3.4	제안방법의 흐름도	21
그림 3.5	탐색범위의 $R_{s,max}$ 와 $G_{sh,max}$ 의 정의	23
그림 4.1	제안방법의 최적화의 진행과정	25
그림 4.2	특성곡선 추출을 위한 PSIM 회로	26
그림 4.3	I-V 특성곡선	28
그림 4.3	I-V 특성곡선 (계속)	29
그림 4.4	P-V 특성곡선	30
그림 4.4	P-V 특성곡선 (계속)	31
그림 4.5	알고리즘의 성능비교	32
그림 4.5	알고리즘의 성능비교 (계속)	33
그림 4.6	PSIM의 태양광 패널 모델	37
그림 4.7	PSIM 모델의 동작원리	39
그림 4.8	PSIM의 태양광 모듈 프로그램	40
그림 4.9	제안모델의 C block 구현	42
그림 4.9	제안모델의 C block 구현 (계속)	43
그림 4.10	제안모델의 동작원리	44

그림 4.11 제안모델의 소프트웨어 구현	45
그림 4.12 제안모델의 출력특성 결과 (KC200GT)	45
그림 4.13 제안모델과 PSIM 모델의 온도영향 비교	47
그림 4.13 제안모델과 PSIM 모델의 온도영향 비교 (계속)	48
그림 4.14 제안모델과 PSIM 모델의 일사량영향 비교	49
그림 4.14 제안모델과 PSIM 모델의 일사량영향 비교 (계속)	50
그림 4.15 하드웨어 회로의 개념도	51
그림 4.16 태양광 시뮬레이터의 동작범위	53
그림 4.17 전류제어의 제어흐름도	54
그림 4.18 점근적 전달함수	56
그림 4.19 Type3 제어기	57
그림 4.20 Type2 제어기	58
그림 4.21 MATLAB 검증	59
그림 4.22 Type2 제어기의 구성	61
그림 4.23 제안모델의 하드웨어 시뮬레이션 회로	62
그림 4.24 기준지령에 대한 동적응답	63
그림 4.25 부하외란에 대한 동적응답	64
그림 4.26 KC125GT의 일사량에 대한 I-V 및 P-V 특성곡선	65
그림 4.27 KC125GT의 온도에 대한 I-V 및 P-V 특성곡선	66
그림 4.28 제안모델의 일사량에 대한 응답특성	67
그림 4.29 제안모델의 온도에 대한 응답특성 (25 to 50 °C)	68

표 목 차

표 2.1 데이터 시트에서 제공하는 패널정보 (KC125GT)	8
표 4.1 태양광 패널의 데이터 시트 정보	27
표 4.2 각 방법의 통계적 비교검증	34
표 4.3 함수모델의 입력 파라미터	37
표 4.4 물리모델의 입력 파라미터	38
표 4.5 벽-컨버터의 설계사양	52
표 4.6 제어기의 설계사양	60

1. 서론

1.1 연구배경 및 필요성

최근 화석연료의 사용 및 고갈에 따른 지구의 환경문제가 대두됨에 따라서 신재생 에너지에 대한 관심이 크게 증가하고 있는 추세이다. 그 중에서도 태양광 (photovoltaic, PV) 발전 시스템은 무한한 에너지원인 태양으로부터 에너지를 얻을 수 있고 친환경적이고 무해하다는 장점을 부각시켜 최근 많이 연구되고 사용처가 증가하고 있다. 이러한 태양광 발전 시스템의 주요 관심사 중 하나는 ‘제한된 사용시간인 낮 시간 동안에 태양으로부터 어떻게 최대의 에너지를 얻을 것인가’이다. 이를 위해서 태양광 패널 자체의 변환효율을 증가시켜서 보다 많은 에너지를 태양으로 얻어내거나 에너지 변환장치의 효율을 최대한으로 증가시켜서 시스템을 효율적으로 사용하고자 하는 많은 노력이 있다.

태양광 시스템의 개발은 태양광 패널의 특성을 정확하게 파악할 필요가 있으나 실제 태양광 패널의 출력은 주변의 온도 및 일사량의 변화에 따라서 실시간으로 변하여 그 평가가 매우 어렵다. 그래서 일반적으로 PCS (power conditioning system)의 개발과정에서는 사용자가 요구하는 다양한 개발환경에 따라서 출력 및 특성을 쉽게 제공할 수 있는 태양광 시뮬레이터가 많이 사용되고 있다. 보통 태양광 시뮬레이터의 경우 다이오드 기반의 등가 회로 모델을 사용하고 있으며, 다이오드 기반의 회로는 태양광 패널의 물리적 특성을 간단하고 쉽게 표현할 수 있다. 이때, 사용자는 다이오드 기반의 회로의 주요 파라미터의 값을 정해서 회로의 특성을 사용자의 요구에 맞게 변화시키면서 사용을 할 수 있다. 그러나 주요 파라미터는 비선형적인 특성을 포함하고 있기 때문에 태양광 패널의 특징을 직접적으로 이용해서 파라미터를 얻는 것은 매우 어려운 일이다. 따라서 비선형적 특성으로부터 주요 파라미터를 얻기 위해서는 다양한 종류의 전기적 모델링 방법이 필요하다.[1]

대부분의 모델링 방법은 수치해석적인 방법을 이용해서 파라미터를 추출하는데 이는 비선형적 특성으로 인한 것이다. 그리고 태양광 시스템의 경우 통상적으로 시스템의 에너지 이용을 극대화하기 위해서 최대전력지점추종 (maximum power point tracking, MPPT) 알고리즘을 함께 사용하는데, 이는 최대전력지점 (maximum power point, MPP)에서의 정확도가 중요함을 의미한다. 그리고 실제 태양광 패널이 존재하지 않아도 시뮬레이터만으로 시스템을 구현하고 실험을 하기 위해서 다양한 조건에서의 태양광 패널의 특성을 최소한의 과정을 통해서 빠르고 정확하게 제공하는 것이 중요하다.

1.2 태양광 패널의 특성과 등가모델

태양광 패널의 에너지 발생은 태양광 셀이 태양빛을 받으면 P-N 접합의 광전효과에 의한 전자의 이동을 통해서 전류가 발생하는 원리를 이용한 기술이다. 이러한 현상은 주로 하나의 전류원과 병렬로 연결되어 있는 다이오드를 가지는 이상적인 다이오드 모델로 설명이 가능하며, 이는 물리적 특성 또한 충분히 내포하고 있다. 여기서 병렬로 연결된 다이오드가 한 개일 경우 싱글-다이오드 회로라 하고 두 개면 더블-다이오드 회로라고 한다. 하지만 싱글-다이오드 회로만으로도 태양광 패널의 물리적 특성을 충분히 반영하고 특성을 묘사하는데 문제가 없어 일반적으로 본 논문에서는 싱글-다이오드 모델을 이용한 태양광 패널의 등가회로 모델링을 연구하였다.

그리고 실제 태양광 패널은 많은 수의 태양광 셀을 직렬 혹은 병렬로 연결해서 구성된다는 점을 고려하면, 실제 싱글-다이오드 등가회로는 그림 1.1과 같이 표현할 수 있다. 그림은 전류원 (photovoltaic current, I_{ph})과 다이오드에 두 개의 직병렬 저항이 포함된 형태를 보이며, 이때 직렬저항 (series resistor, R_s)은 태양광 패널에 부착된 각각의 태양광 셀 간의 접촉저항을 의미하고 병렬컨덕턴스 (shunt resistor, G_{sh})는 누설로서 새어나가는 전류를 표현하기 위해서 사용된다. 그리고 회로에서 암전류 (dark current, I_0)와 다이오드 이상계수 (diode ideality factor, A)는 다이오드와 관련된 성분을 의미한다.[2][3]

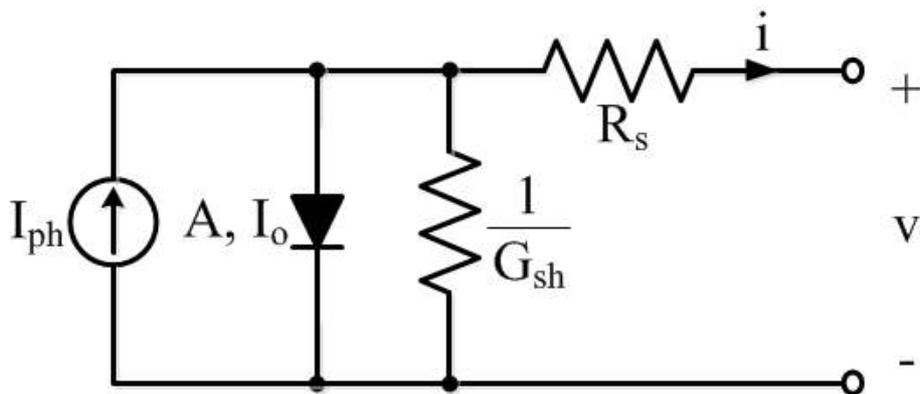


그림 1.1 싱글 다이오드 등가모델

Fig. 1.1 Single-diode equivalent model

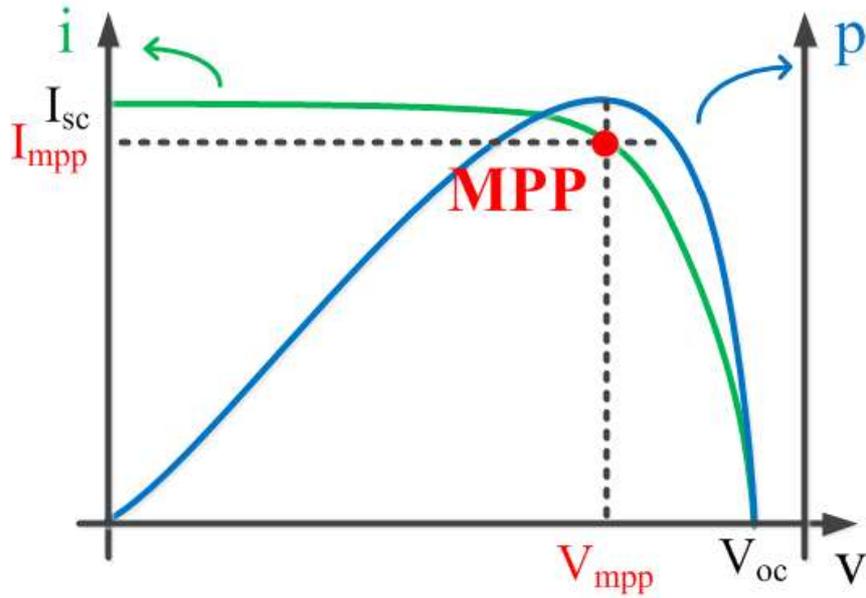


그림 1.2 태양광 패널의 출력특성

Fig. 1.2 Output characteristic of PV panel

그림 1.1의 회로를 이용하면 태양광 패널의 비선형적인 출력특성을 모사할 수 있으며, 이때 출력특성은 그림 1.2와 같이 나타난다. 그림에서 최대전력전류 (maximum power current, I_{mpp})와 최대전력전압 (maximum power voltage, V_{mpp})이 만나는 지점은 태양광 패널의 MPP로 태양광 시스템은 주로 MPPT 방법을 이용해서 MPP 부근에서 운전을 하여 최대의 에너지를 얻는 것을 목표로 한다. 그림 1.2의 비선형적인 태양광 패널의 출력특성은 물리적인 특성을 고려하여 수식으로 표현할 수 있는데 이는 식 (1.1)과 같이 표현할 수 있다.

$$i = I_{ph} - I_o \left(e^{\frac{v + iR_s}{N_s A V_T}} - 1 \right) - (v + iR_s) G_{sh} \quad (1.1)$$

여기서 V_T 는 아래의 수식으로 정의 된다.

$$V_T = \frac{kT}{q} \quad (1.2)$$

이때, 각 파라미터는 아래와 같이 정의한다.

- I_{ph} : 광전류 (A)
- I_o : 암전류 (A)
- R_s : 직렬저항 (Ω)
- G_{sh} : 병렬컨덕턴스 (S)
- N_s : 직렬연결된 PV 셀의 수
- A : 다이오드 이상계수 (실리콘 : 1~2)
- V_T : 열전압 (V)
- k : 볼츠만 상수 (1.602×10^{-19} J/K)
- q : 전자의 전하량 (1.38×10^{-23} C)
- T : 절대온도 (K)

1.3 연구목표 및 내용

각각의 파라미터 중 R_s , G_{sh} , A , I_{ph} , I_o 는 싱글-다이오드 등가회로를 구성하는 미지수로서 패널의 종류에 따라서 사용자가 설정을 해줄 필요가 있는 값이다. 하지만 5개의 미지수를 구하기 위해서 비선형적인 태양광 패널의 출력특성을 분석할 필요가 있는데 이를 직접적인 측정을 통해서 얻는 것이 매우 어렵다. 그러므로 다양한 종류의 모델링 방법을 통해서 미지의 파라미터를 구하는 것이 필요하다.

태양광 패널의 전기적 등가회로를 얻기 위해서 사용하는 모델링 방법 중에서 가장 보편적인 방법은 실제 태양광 패널이 없이 제조사가 제공하는 데이터시트만으로도 모델링이 가능한 데이터시트 기반의 방법이다. 이 방법은 데이터시트만 있으면 빠르고 쉽게 태양광 패널의 등가회로를 구성할 수 있고 실제 패널이 없어도 태양광 패널의 성능평가나 전력변환기의 효율개선과 같은 하드웨어 실험을 진행할 수 있게 도와준다는 장점을 가지고 있다. 이에 따라서 많은 데이터시트 기반의 태양광 패널 모델링 방법이 연구되어 왔다.

본 논문은 다음과 같은 순서로 구성되어 있다.

1장은 서론으로 본 논문에서 제안하는 데이터시트 기반의 태양광 패널의 전기적 등가회로 모델링방법 및 모델의 연구배경과 필요성에 대해서 설명하였다.

2장에서는 데이터시트 기반의 모델링에 대해서 장점과 최근의 주요이슈에 대해서 정리하고 이 방법에서 마주할 수 있는 데이터시트의 정보부족 문제에 대해서 설명한다. 그리고 앞서 연구된 다양한 기존의 태양광 패널의 모델링 방법을 3개의 대표적인 범주로 분류하고 각 범주에서 발견할 수 있는 장점 및 단점을 분석하여 필요한 개선점을 제시한다.

3장에서는 기존의 모델링 방법의 문제점을 극복하기 위해서 본 논문에서 제안하는 새로운 태양광 패널의 전기적 등가회로 모델링 방법에 대해서 기술하며, 이와 관련된 수식 유도 과정 및 최적화 방법에 대한 이론을 설명하였다. 그리고 추가적으로 최적화를 보다 적절하게 사용하기 위한 해의 탐색범위 및 초기값을 제안하였다.

4장은 제안모델의 검증 및 소프트웨어 구현과 하드웨어 구현에 대해서 설명한다. 이때, 모델은 PSIM과 MATLAB과 같은 시뮬레이션 프로그램을 통해서 구현하였고 비교를 위해서 EN50530의 유럽규격을 사용해서 제안방법의 성능을 평가하며, 성능은 기존방법과의 비교를 통해서 그 타당성이 검증된다. 그리고 제안모델의 사용성을 확장시키기 위해서 PSIM만으로 제안모델을 구현하며, 이때 PSIM의 C block과 C 언어를 사용하였다. 마찬가지로 제안모델을 하드웨어 회로로서도 사용하기 위해서 하드웨어회로 및 제어를 설계하였고 PSIM을 통해서 시뮬레이션 검증을 하였다.

2. 기존의 데이터시트 기반의 모델링 방법

2.1 데이터시트 기반의 모델링

데이터시트 기반의 태양광 패널 모델링 방법은 패널의 제조사가 제공하는 데이터시트만을 이용해서 태양광 패널의 등가회로 모델을 얻는 방법이다. 실제 태양광 패널의 데이터를 측정하면, 등가회로 모델을 쉽게 얻을 수 있지만, 실제 패널이 없는 경우에는 데이터시트 기반의 방법이 매우 유용하다. 하지만 모델링 과정에서 데이터시트만으로는 충분한 정보를 얻는 것이 어려워 각각의 기존방법에 따라서 모델링을 하는 방법이 다르고 이는 오차의 원인이 되기도 하므로 기존의 모델링 방법에 대한 분석을 2장에 정리하였다.

2.1.1 데이터시트 기반 모델링의 장점과 주요이슈

실제 태양광 패널은 온도와 일사량의 변화에 따라서 출력특성이 항상 변하기 때문에 태양광 PCS의 개발과정에서 실제 패널을 사용하기란 매우 어려운 일이다. 그래서 일반적으로 개발과정에서 회로 시뮬레이션을 많이 사용하게 되는데 이때 필요한 것이 태양광 패널의 회로적인 특성을 표현할 수 있는 등가회로 모델이다. 등가회로 모델링을 위한 다양한 방법이 존재하지만, 그 중에서 실제 패널의 데이터를 측정한 뒤 이를 이용해서 모델링을 하는 방법과 실제 패널이 없어도 모델링이 가능한 데이터시트 기반의 방법이 대표적이다. 이때, 데이터시트 기반의 방법은 실제 태양광 패널이 없어도 등가회로 모델을 쉽고 빠르게 구할 수 있다는 장점을 가진다.

하지만 데이터시트로부터 얻을 수 있는 정보는 한정적이고 부족하기 때문에 데이터시트 기반의 방법의 주요이슈는 ‘한정된 정보 내에서 어떻게 정확한 모델을 얻을 것인지’이다. 이를 위해서 다양한 종류의 방법이 제안된 바 있지만, 대부분의 방법에서의 공통점은 제한된 정보량을 극복하기 위해서 수치해석적인 방법을 사용한다는 점이다. 본 장에서는 데이터시트 기반의 모델링 문헌들의 특징을 구분하였고 대표적으로 3개의 방법으로 분류한 뒤 분석을 하였다.[4]-[12]

2.1.2 데이터시트의 정보부족 문제

태양광 패널은 일반적으로 싱글-다이오드 등가회로로 표현이 가능한데 등가회로의 5개의 파라미터 R_s , G_{sh} , A , I_{ph} , I_o 를 사용자가 필요로 하는 패널의 특징에 맞게 결정한 뒤에 사용

해야 한다. 여기서 파라미터를 결정하기 위해서 전기적인 모델링 방법이 필요하며, 이는 데이터시트 기반의 모델링 방법을 이용해서 극복할 수 있다.

기본적으로 데이터시트로부터 사용자가 얻을 수 있는 정보는 3개의 특성지점에 관한 정보이며, 이는 그림 2.1에서 나타나 있듯이 MPP, I_{sc} , V_{oc} 의 3개의 지점이다. 그리고 태양광 패널의 특성식인 식 (1.1)을 이용해서 MPP에서의 기울기에 대한 정보를 얻을 수 있다. 이를 정리하면 아래와 같다.

- 1) 최대 전력지점 (maximum power point, MPP).
- 2) 최대 전력지점에서의 기울기는 0.
- 3) 개방전압 지점 (open circuit voltage, V_{oc}).
- 4) 단락전류 지점 (short circuit current, I_{sc}).

이때, 3개의 특성지점에 관한 정보는 패널의 최대전력(maximum power, P_{max})과 함께 표 2.1과 같이 데이터시트에서 확인할 수 있으며, 표 2.1의 데이터는 Kyocera社에서 제공하는 KC125GT 패널의 실제 데이터시트의 정보이다. 하지만 표에서 제공하는 정보들은 모두 표준 테스트 조건 (standard test condition, STC)에서만 성립하는 값이며, 태양광 패널의 주변

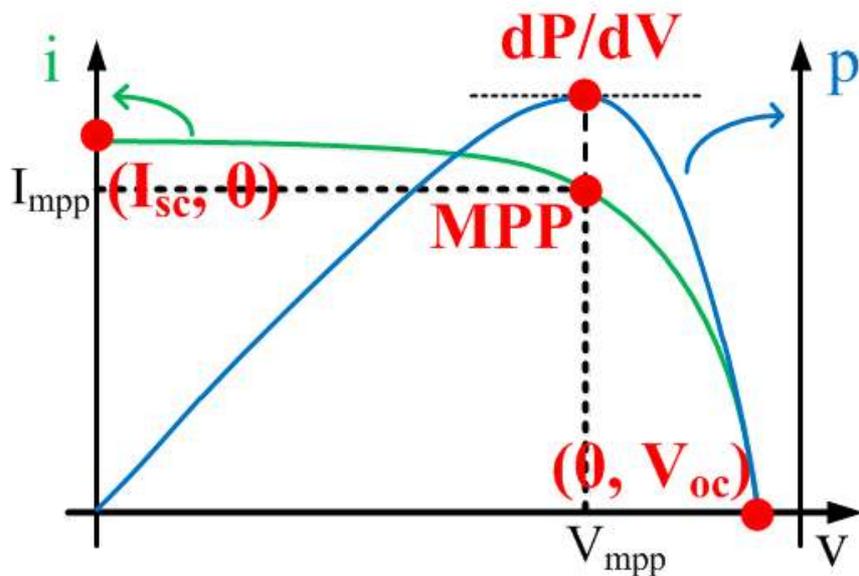


그림 2.1 태양광 패널의 특성지점

Fig. 2.1 Characteristic point of PV panel

표 2.1 데이터 시트에서 제공하는 패널정보 (KC125GT)

Table. 2.1 Specifications for PV panel (KC125GT)

Electrical Performance under Standard Test Condition (STC)	
Maximum Power (Pmax)	125 W (Tolerance: +10% / -5%)
Maximum Power Voltage (Vmpp)	17.4 V
Maximum Power Current (Impp)	7.20 A
Open Circuit Voltage (Voc)	21.7 V
Short circuit Current (Isc)	8.00 A

온도 및 일사량이 변할 경우에 대해서는 정보를 제공하고 있지 않다. 결국, 사용자는 STC에서의 데이터만을 가지고 태양광 패널의 등가회로를 모델링해야 한다. 하지만 싱글-다이오드 등가회로가 요구하는 파라미터는 기본적으로 R_s , G_{sh} , A , I_{ph} , I_o 의 5개로 여기서 그림 2.2와 같이 데이터시트 기반의 모델링의 파라미터를 추출하기 위한 데이터시트의 특성정보가 부족하다는 문제점이 드러난다. 결국, 태양광 패널의 제조사가 제공하는 4개의 조건을 이용해서 5개의 미지수를 한 번에 구하는 것은 어렵기 때문에 기존의 모델링은 다양한 방법을 통해서 정보부족 문제를 해결하고 있다.

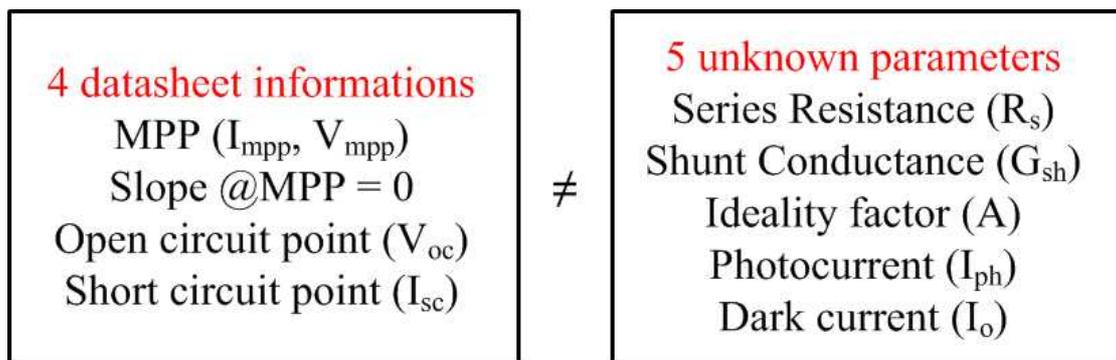


그림 2.2 데이터시트의 정보부족 문제

Fig. 2.2 Lack of information in datasheet

2.2 기존모델링 분석

2.2.1 파라미터 제거법

데이터시트의 정보가 부족한 문제를 해결하기 위해서 기본적으로 데이터시트 기반의 방법들은 수치해석적인 방법을 채택하고 있다. 하지만 가장 기본적인 해결방법으로 그림 2.3 과 같이 전기적 등가회로의 파라미터의 수를 하나 줄여서 미지수와 데이터시트 정보의 수를 맞춰 주는 방법이 존재한다.[4]-[7] 파라미터를 제거하기 위해서 직렬저항 R_s 의 경우 매우 작다고 가정하거나, 병렬컨덕턴스 G_{sh} 는 매우 작다고 가정하면 회로 내에서 각각의 파라미터를 생략할 수 있는데 이 방법의 장점은 회로에서 파라미터를 하나 제거함으로써 미지수의 수를 데이터시트의 정보와 같게 4개로 맞춰 주었다는 것이다. 따라서 미지수와 정보의 수가 같기 때문에 회로의 파라미터는 수치해석적인 반복적 과정이 필요 없이 연립방정식을 사용하는 것만으로도 구할 수 있다. 이는 데이터 시트 기반의 파라미터 추출방법의 장점인 빠른 전기적 등가회로 모델을 제공한다는 점에서 우수함을 가지는 방법이다. 하지만 회로의 물리적 특성을 표현하는 파라미터를 하나 제거함으로써 회로의 물리적인 표현력은 감소하여 태양광 패널의 출력을 정확하게 표현한다는 장점은 사라지고 경우에 따라서는 오차가 심하고 부정확한 출력특성을 제공할 수 있다.

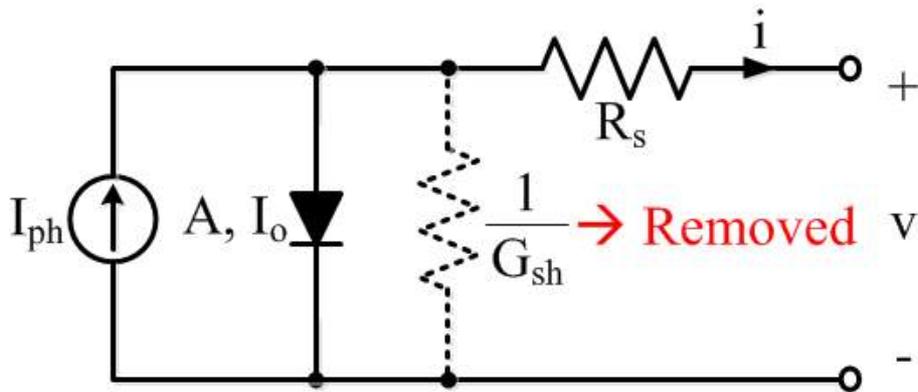


그림 2.3 파라미터 제거방법

Fig. 2.3 A method reducing one of the model parameters

2.2.2 파라미터 고정법

파라미터를 생략하는 것을 통해서 파라미터의 추출속도는 향상되지만 정확도가 떨어지는 경우가 발생한다. 이를 극복하기 위해서 다른 문헌의 경우에는 파라미터 하나의 값을 사용자가 임의로 고정시킨 뒤에 남은 4개의 미지수를 4개의 방정식을 통해서 구하는 방법을 제안하고 있다.[8][9] 대표적인 방법으로 다이오드의 이상계수를 고정값으로 가정하고 나머지 파라미터를 구하는 방법이 존재하는데 일반적으로 태양광 패널의 재료가 실리콘임을 생각해서 다이오드의 이상계수는 1~2 사이의 값으로 정할 수가 있다는 점을 이용한 것이다. 이는 그림 2.4과 같이 표현이 가능하며, 남은 4개의 파라미터는 4개의 방정식을 이용한 반복적인 과정을 통해서 최적의 해를 찾아나간다. 이 방법은 MPP의 정확도가 좋은 태양광 패널의 모델을 얻을 수 있다는 장점이 있지만, 그림 2.5에서 볼 수 있듯이 경우에 따라서는 MPP의 전압이 어긋나는 경우가 발생할 수 있다는 단점도 가진다. 그림 2.5는 파라미터 고정법 중에서 Villalva가 제안한 방법으로 이 방법은 A를 고정시킨 후에 나머지 4개의 파라미터를 찾는 방법이다. 여기서 A는 실리콘을 재료로 하는 태양광 패널의 경우 일반적으로 1에서 1.5사이의 값으로 결정할 수 있다. 그림에서는 A를 양 끝 값으로 결정한 뒤에 각 I-V 특성곡선의 MPP를 표시한 것이다. 확대한 그림에서 볼 수 있듯이 등가모델의 MPP는 비슷하지만, 사용자가 결정하는 A의 값에 따라서 V_{mpp} 가 조금씩 달라지는 것을 확인할 수 있으며, 이는 결국 모델의 오차로서 작용을 하게 될 것이다.

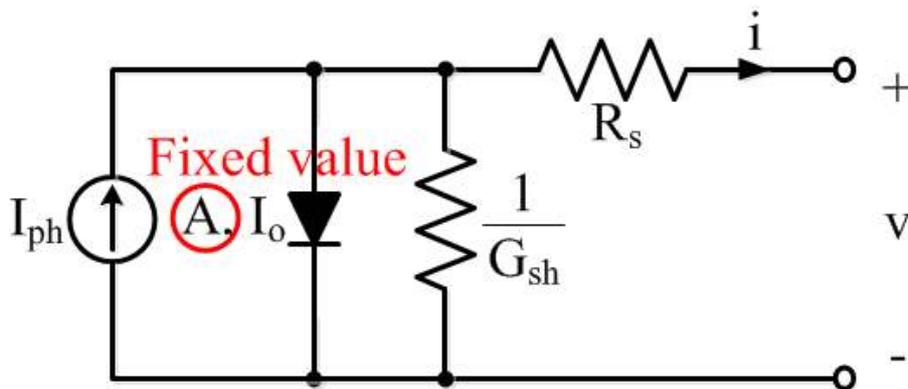


그림 2.4 파라미터 고정방법

Fig. 2.4 A method fixing one of the model parameters

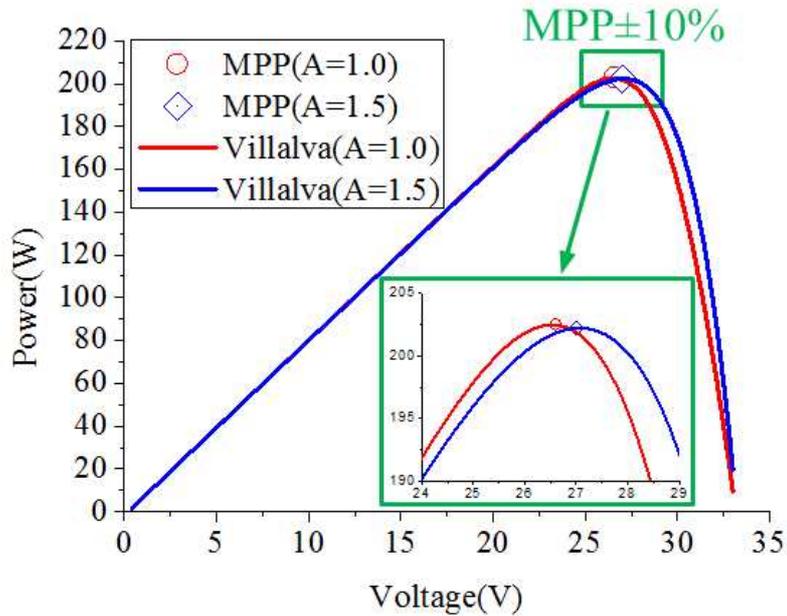


그림 2.5 파라미터 고정방법의 단점

Fig. 2.5 Disadvantage of the parameter-fixing method

2.2.3 기울기 조건 추가법

마지막으로 또 다른 문헌은 추가적인 조건을 사용해서 파라미터를 추출하는 방법을 제안하였다[10]-[12]. 이 방법은 추가적인 조건으로 방정식의 수를 5개로 맞추는 것으로 미지수와 방정식의 수가 불일치하는 문제를 극복한다. 다만, 추가적인 조건이 추가됨에 따라서 최적의 파라미터를 찾기 위해서는 더 많은 반복과정이 필요하다. 추가적인 조건은 그림 2.6에서 확인 할 수 있다. 그림에서 추가조건은 단락전류와 개방전압의 기울기 식을 통해서 얻을 수 있는데, 이때 근사식을 통해서 수식을 유도한다. 이 방법은 추가적인 조건을 통해서 모델의 정확도를 향상시킬 수 있다는 장점을 가지지만, 조건이 추가된 만큼 파라미터의 추출을 위한 계산시간이 증가한다는 단점도 존재한다. 그리고 근사식을 통해 추가조건을 유도하기 때문에 이는 모델의 오차에도 영향을 준다. 그림 2.7은 파라미터의 추출과정에서 추가조건을 만족하지 못하는 경우 발생할 수 있는 오차에 대해서 확인할 수 있는 그림으로 기울기 조건을 추가하는 방법 중 하나인 Pedro방법을 구현한 뒤 그 결과를 추출한 것이다. Pedro방법은 I_{sc} 에서의 기울기 조건을 추가해서 등가회로를 모델링하는 방법으로 그림은 근사식으로 얻어진 I_{sc} 에서의 기울기가 실제 태양광 패널에서 다를 경우 발생할

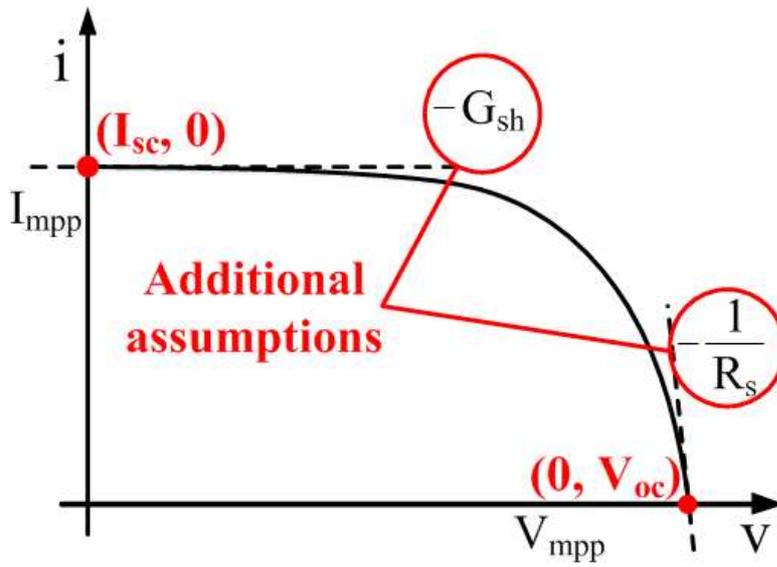


그림 2.6 파라미터 추출을 위한 추가조건

Fig. 2.6 Additional conditions for parameter extraction

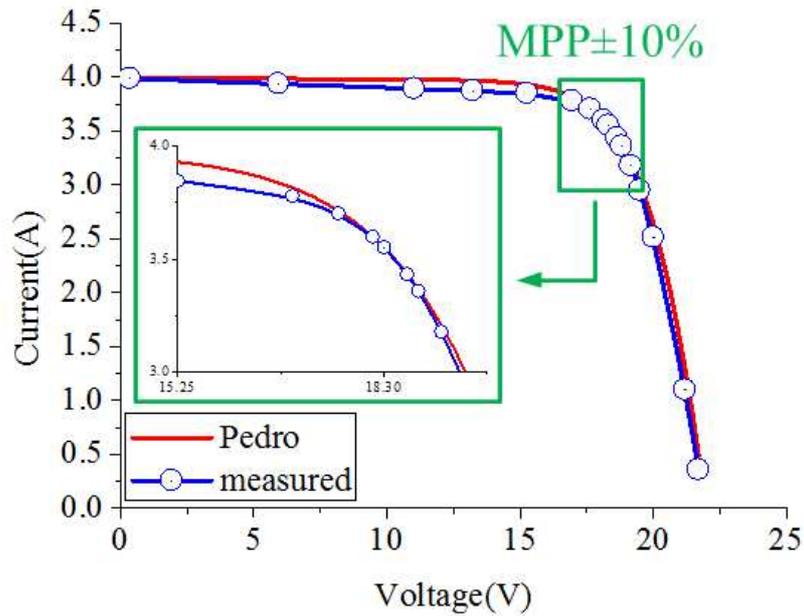


그림 2.7 조건 추가법의 단점

Fig. 2.7 Disadvantage of the additional condition method

수 있는 오차에 대해서 설명하고 있다. 확대한 부분을 보면, **MPP** 부근에서 오차가 발생할 수 있음을 확인할 수 있다.

현재까지 앞서 설명한 기존의 대표적인 논문의 방법들이 5개의 미지수와 4개의 방정식의 수를 맞추기 위해서 진행한 내용을 살펴보았다. 결국, 태양광 패널의 전기적 등가회로를 모델링하기 위해서는 미지수와 방정식의 수의 불일치 문제를 해결하는 것이 가장 큰 문제이며, 추가적으로 정확도와 빠른 추출속도를 제공해야 함을 확인할 수 있었다. 이에 따라서 다음 3장에서는 제안방법을 통해서 기존의 방법이 해결하지 못한 문제를 해결하도록 한다.

3. 제안하는 데이터시트 기반의 방법

제안하는 태양광 패널의 등가회로 모델링 방법은 미지수와 방정식의 수의 불일치 문제를 극복하기 위해서 최적화 방법을 통해서 파라미터를 추출하는 것이다. 선행문헌[13]-[16]에서도 최적화 방법을 사용해서 태양광 패널의 모델링을 하는 방법이 제시된 바 있으며, 최적화 방법은 전기적 등가회로의 파라미터를 제거하거나 고정하지도 않으며, 추가적인 조건을 필요로 하지 않는 방법으로 수식의 유도과정에서 근사식을 최소화 하며, 최적화를 이용하므로 모델의 정확도를 보장하는 방법이다. 하지만 기존의 최적화 방법을 사용하는 모델링 방법의 경우 실제 태양광 패널의 출력을 측정해야 하거나 추가적인 최적화 프로그램이 필요한 방법으로 사용하기에 불편한 점이 존재한다. 그래서 제안방법은 데이터시트에서 제공하는 4가지의 조건만 있으면 태양광 패널의 전기적 등가회로를 구할 수 있는 방법을 제안하며, 그 개념도는 그림 3.1과 같이 나타낼 수 있다. 그림에서 확인 할 수 있듯이 제안 방법은 4개의 데이터 시트로부터 얻을 수 있는 조건 중에서 2개의 최대 전력지점의 조건 1)과 2)를 사용해 3개의 미지수 R_s , G_{sh} , A 를 구하며, 나머지 2개의 양 끝 동작지점 조건 3)과 4)로 남은 미지수인 I_{ph} 와 I_o 를 구한다. 결국 데이터시트의 정보부족 문제를 극복하기 위

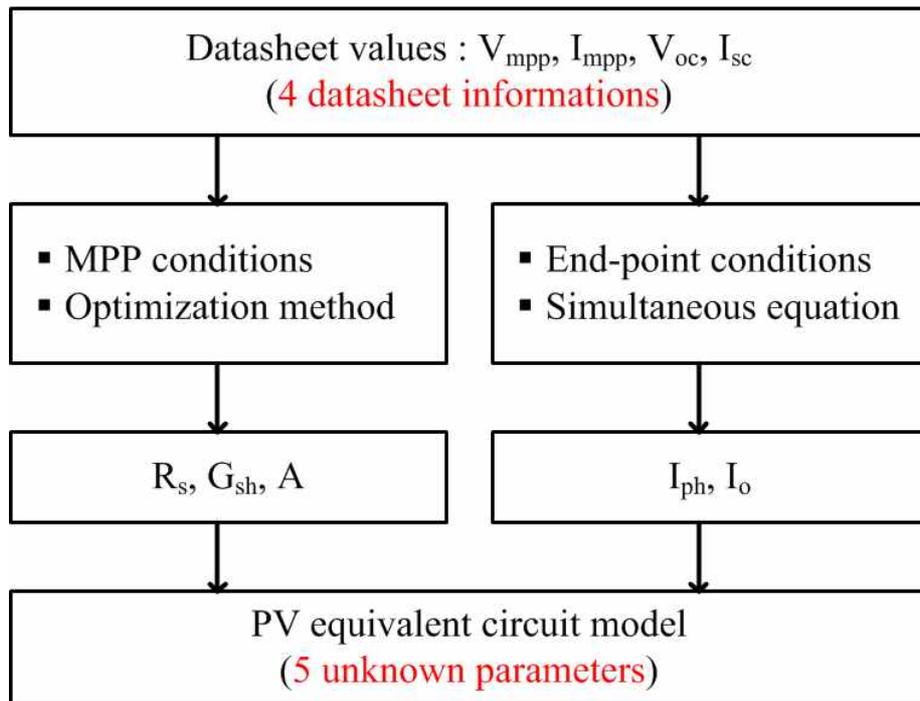


그림 3.1 제안방법의 개념도

Fig. 3.1 Overview of the proposed method

해서 제안방법은 최적화 방법을 채택하였고 결과적으로 5개의 미지수를 구할 수 있게 된다.

3.1 제안방법의 수식유도

우선, 최적화를 사용하는 제안방법은 목표함수의 설정이 필요하다.[13]-[16] 목표함수는 태양광 패널의 기본 수식인 식 (2.1)로부터 얻을 수 있으며, 이 식에 앞서 제시한 4개의 동작조건인 1) 최대 전력지점, 2) 최대 전력지점에서의 기울기는 0, 3) 개방전압 지점, 4) 단락전류 지점을 이용해서 구할 수 있다. 여기서 일반적인 태양광 시스템이 MPPT 알고리즘과 같이 사용되는 것을 고려하면 최대 전력지점에서의 기울기가 더 중요하다고 볼 수 있다. 그래서 제안방법은 최대 전력지점 조건인 1)과 2)의 조건만을 이용해서 목표함수를 설정하였다.

3.1.1 목표함수 설정

목표함수를 설정하기 위해서 식 (2.1)에 1) 최대 전력지점의 조건을 대입하면 아래와 같은 수식을 얻을 수 있다.

$$I_{mpp} = I_{ph} - I_0 e^{\frac{V_{mpp} + I_{mpp}R_s}{N_s A V_T}} - (V_{mpp} + I_{mpp}R_s)G_{sh} \quad (3.1)$$

이 수식을 통해서 최대 전력지점과 관련된 수식을 얻을 수 있으며, 식 (3.1)은 음함수의 형태인 식 (3.2)로 바꿀 수 있다.

$$f(R_s, G_{sh}, A) - I_{mpp} = 0 \quad (3.2)$$

위의 음함수 식은 목표함수 설정을 위한 첫 번째 수식이 된다.

다음으로 목표함수를 설정하기 위한 두 번째 수식을 얻기 위해서 태양광 패널의 출력 전력의 수식을 구하면 아래와 같다.

$$p(v) = iv \quad (3.3)$$

위의 수식을 최대 전력지점에서의 기울기를 구하기 위해서 미분공식을 적용하면 식 (3.4)와 같이 표현된다.

$$\frac{dp}{dv} = \frac{d(vi)}{dv} = i + \frac{di}{dv}v \quad (3.4)$$

그리고 식 (2.1)의 음함수 형태는 다음과 같으며,

$$i = f(i, v) \quad (3.5)$$

위의 음함수를 미분하면,

$$di = di \frac{\partial f(i, v)}{\partial i} + dv \frac{\partial f(i, v)}{\partial v} \quad (3.6)$$

식 (3.6)을 얻을 수 있다. 따라서

$$\frac{di}{dv} = \frac{\frac{\partial}{\partial v} f(i, v)}{1 - \frac{\partial}{\partial i} f(i, v)} \quad (3.7)$$

을 얻을 수 있다. 이때, 식 (3.4)와 (3.7)을 이용한 수식에 조건 1) 최대 전력지점을 대입하면, 2)의 조건인 최대 전력지점에서의 기울기는 0 이라는 식 (3.8)을 얻을 수 있다.

$$\left. \frac{dp}{dv} \right|_{@MPP} = I_{mpp} - V_{mpp} \frac{G_{sh} \left(\frac{I_{sc}/G_{sh} - V_{oc} + I_{sc}R_s}{N_s A V_T} e^{\frac{V_{mpp} + I_{mpp}R_s - V_{oc}}{N_s A V_T}} + 1 \right)}{1 + R_s G_{sh} \left(\frac{I_{sc}/G_{sh} - V_{oc} + I_{sc}R_s}{N_s A V_T} e^{\frac{V_{mpp} + I_{mpp}R_s - V_{oc}}{N_s A V_T}} + 1 \right)} = 0 \quad (3.8)$$

위의 수식은 바로 음함수의 형태인 식 (3.9)으로 표현이 가능하며,

$$g(R_s, G_{sh}, A) = 0 \quad (3.9)$$

제안방법은 식 (3.2)와 (3.9)를 이용해서 아래의 목표함수를 설정할 수 있다.

$$E(R_s, G_{sh}, A) \equiv (f(R_s, G_{sh}, A) - I_{mpp})^2 + g^2(R_s, G_{sh}, A) \quad (3.10)$$

식 (3.10)으로 구해진 목표함수를 이용해서 제안방법은 R_s , G_{sh} , A 의 값을 파라미터의 제거나 고정을 비롯한 추가적인 조건도 없이 한 번에 구할 수 있게 된다. 이것은 2개의 조건만으로 3개의 미지수를 구한 것을 의미하며, 남은 2개의 조건을 통해서 나머지 2개의 미지수인 I_{ph} 와 I_o 를 쉽게 구할 수 있음을 의미한다.

3.1.2 I_{ph} 및 I_o 수식 유도

앞서 설명한 것과 마찬가지로 I_{ph} 와 I_o 는 남은 두 개의 조건인 3)과 4)를 이용해서 구할 수 있다. 먼저, 식 (2.1)로부터 조건 3) 개방전압 지점을 고려하면,

$$I_{ph} = I_o e^{\frac{V_{oc}}{N_s A V_T}} + V_{oc} G_{sh} \quad (3.11)$$

를 얻을 수 있다. 식 (3.11)로부터 I_{ph} 와 관련된 수식이 바로 얻어지는 것을 확인 할 수 있으며, 마찬가지로 식 (2.1)에 조건 4) 단락전류 지점을 고려하면, 아래의 식을 얻을 수 있다.

$$I_{sc} = I_{ph} - I_o e^{\frac{I_{sc} R_s}{N_s A V_T}} - I_{sc} R_s G_{sh} \quad (3.12)$$

이때, I_o 에 대한 식을 얻기 위해서 식 (3.11)을 식 (3.12)에 대입하면,

$$I_{sc} = I_o \left(e^{\frac{I_{sc} R_s}{N_s A V_T}} - e^{\frac{V_{oc}}{N_s A V_T}} \right) + (V_{oc} - I_{sc} R_s) G_{sh} \quad (3.13)$$

을 얻을 수 있다. 식 (3.13)을 I_o 에 대해서 정리하면,

$$I_o = \frac{I_{sc} - (V_{oc} - I_{sc} R_s) G_{sh}}{e^{\frac{I_{sc} R_s}{N_s A V_T}} - e^{\frac{V_{oc}}{N_s A V_T}}} \quad (3.14)$$

를 얻을 수 있고 아래의 조건을 적용하면,

$$V_{oc} \gg I_{sc}R_s \quad (3.15)$$

최종적인 I_o 식의 형태는 다음과 같다.

$$I_o = [I_{sc} - (V_{oc} - I_{sc}R_s)G_{sh}]e^{-\frac{V_{oc}}{N_sAV_T}} \quad (3.16)$$

여기서 I_{ph} 와 I_o 는 R_s , G_{sh} , A 의 파라미터 값에 영향을 받아서 값이 변한다. 결국, 식 (3.10), (3.11) 그리고 (3.16)을 최적화 방법과 함께 이용하면 미지수 R_s , G_{sh} , A , I_{ph} , I_o 를 모두 구할 수 있다. 이때, 목표함수를 만족하는 최적의 값을 찾기 위해서 Powell 최적화 방법을 사용한다.

3.2 Powell 최적화 방법

비선형적인 문제를 풀기위한 수많은 수치해석 방법들 중에서 본 논문은 Powell 최적화 방법을 사용하였다. 이 방법은 목표함수인 식 (3.10)을 최소화 하는 값을 찾아 주는데 해의 탐색에서 미분을 사용하지 않는 방법으로 매우 안정적이면서 단순한 형태로 구현이 쉬워 추가적인 최적화 소프트웨어가 필요하지 않는 방법이다. 또한 최적화 방법의 특징상 계산 시간 또한 빠르다는 장점을 가진다[17].

Powell 최적화 방법의 기본적인 동작은 초기값 X_i 로부터 다음값인 $X_{i+1}=X_i+\alpha_iS_i$ 로 진행을 하면서 최적의 값을 찾아가는 방법으로 탐색벡터 S 를 이용해서 각 파라미터 R_s , G_{sh} , A 의 단위벡터의 방향으로 탐색을 하게 된다. 즉, R_s 에 대해서 $S_1 = [1, 0, 0]$, G_{sh} 에 대해서 $S_2 = [0, 1, 0]$, A 에 대해서 $S_3 = [0, 0, 1]$ 이 되는 것이다. 그리고 마지막으로 $S_4 = X_3-X_1$ 의 방향으로 진행을 하는 것으로 하나의 사이클을 마친다. 이러한 1사이클의 과정이 최적화를 찾을 때까지 반복되며, 각 사이클 마다 탐색벡터는 Powell shift를 이용해서 탐색 속도를 가속하게 된다. 이것은 그림 3.2의 수도코드를 통해서 쉽게 표현이 가능하다. 그리고 해를 탐색하는 과정 중에서 α 의 값은 황금분할법을 이용해서 매 사이클 마다 계산이 된다. 이때, α 는 각 축에서 바라본 목표함수의 값을 최소화 하는 값을 의미하며, 본 논문에서는 α 를 구하는 황금분할법의 종료조건은 $\epsilon=1 \times 10^{-3}$ 으로 정하였다. 결국 최종적인 Powell 최적화 방

```

Step 1: Choose initial  $X_1$ ,  $N_c$  (number of cycle)
        Determine the  $\varepsilon$  : tolerance
        Calculate  $f(X_1)$ ;
        Set  $j = 1$ ; (initialize powell cycle)
        For  $i = 1$ ; (initialize univariate cycle)
             $S_i = e_i$  (univariate step)
Step 2: For each cycle  $j$ 
        For  $i = 1:3$ ;
            If ( $j \geq 2$ )  $S_i = S_{i+1}$  --> Powell shift
             $X_{i+1} = X_i + \alpha_i \cdot S_i$ 
            -->  $\alpha$  is determined by minimizing  $f(X_{i+1})$ 
        End (i cycle)
         $S_j = X_4 - X_1$ 
         $X_j = X_4 + \alpha_j \cdot S_j$ 
        Calculate  $f(X_j)$ 
Step 3:  $\Delta f = f(X_{j+1}) - f(X_j)$ 
         $\Delta X = X_{j+1} - X_j$ 
        If  $|\Delta f| \leq \varepsilon$ ; stop
        If  $\Delta X^T \Delta X \leq \varepsilon$ ; stop
        If  $j = N_c$ ; stop
         $X_1 = X_j$ ;
         $j = j + 1$ ;
        Go to Step 2
    
```

그림 3.2 파웰의 최적화 방법의 수도코드

Fig. 3.2 Pseudo code of Powell's optimization

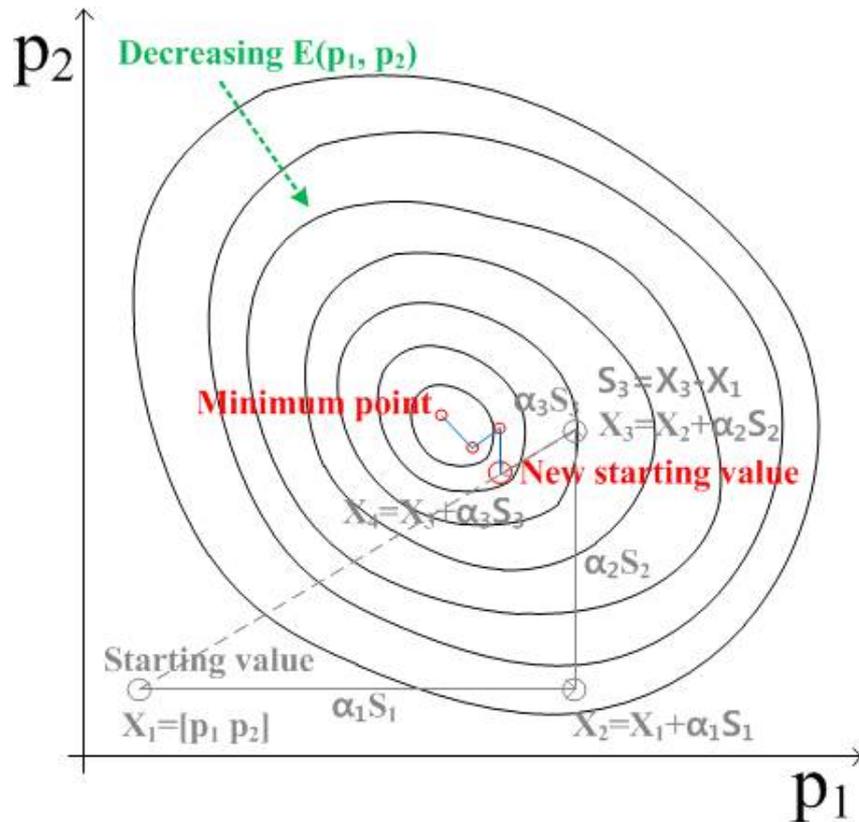


그림 3.3 파워의 최적화 방법의 동작원리

Fig. 3.3 Operating concept of Powell's optimization

법의 동작 원리는 그림 3.3과 같이 나타난다. 그림 3.3는 파워의 방법의 동작원리를 이해하기 쉽게 3변수가 아닌 2변수의 형태로 나타낸 것으로 목표함수가 감소하는 방향으로 해의 탐색방향이 진행하게 된다. 이때, 초기값으로부터 탐색방향이 잘못 설정되는 경우가 발생할 수 있다. 하지만 Powell 방법의 특성상 잘못된 방향의 목표함수는 α 에 의해서 수정되기 때문에 올바른 방향으로 진행하게 되고 결국엔 원하는 최적의 값에 도달하게 된다. 그리고 만약 초기값이 적절하게 설정된다면 해의 수렴속도는 더욱 빨라질 것이다. 결과적으로 목표함수와 파워의 최적화 방법을 모두 포함하는 제안방법의 흐름도는 그림 3.4과 같이 나타낼 수 있다.

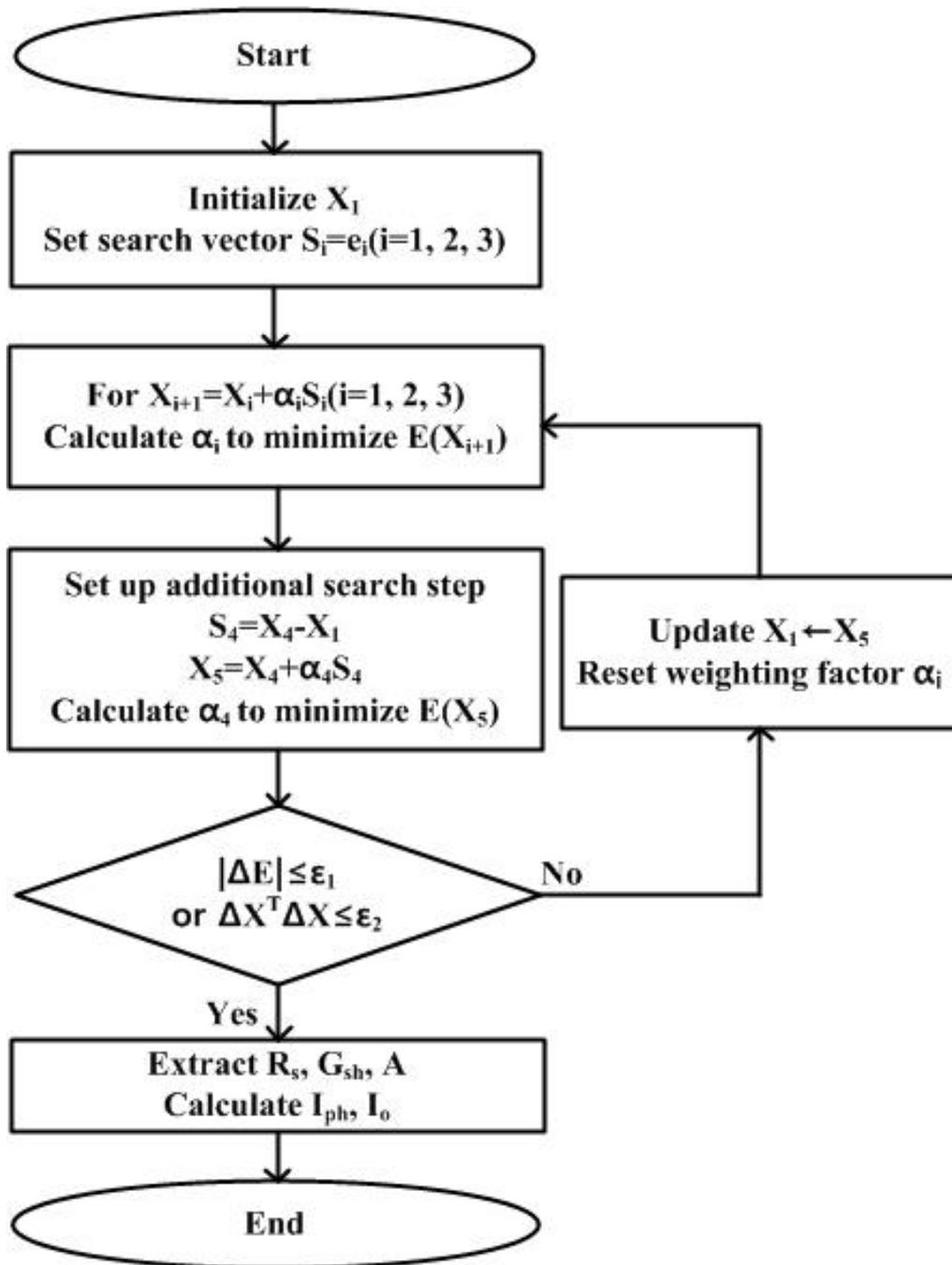


그림 3.4 제안방법의 흐름도

Fig. 3.4 Flow chart of the proposed algorithm

3.3 탐색범위 및 초기값 설정

비선형 문제를 풀기 위한 최적화 방법을 사용하기 위해서는 각 변수에 대한 초기값과 탐색 범위에 대한 설정도 필요하다. 이는 수치해석적인 방법을 사용하기 위해서는 필수적인 것으로 이때 구하고자 하는 각각의 해의 탐색범위를 적절하게 설정해야 한다. 본 논문에서는 태양광 패널의 등가회로의 물리적 의미를 훼손하지 않는 타당한 근거를 가지는 해의 초기값과 탐색범위를 제안하였다.

초기값 및 탐색범위는 데이터시트로부터 등가회로의 물리적 의미를 훼손하지 않는 값으로 쉽게 결정할 수 있는데, 먼저 R_s 의 경우 이상적인 경우 접촉 저항은 존재하지 않는다는 가정을 바탕으로 최소값은 0으로 둘 수 있다. 그리고 최대값의 경우 R_s 가 아무리 크게 변하여도 I-V 커브의 최대 전력지점은 벗어날 수 없다는 점을 고려하면 다음과 같이 결정된다.

$$0 \leq R_s \leq \frac{V_{oc} - V_{mpp}}{I_{mpp}} \quad (3.17)$$

마찬가지로 G_{sh} 의 경우 이상적으로 누설 성분이 존재하지 않는다고 가정하면 최소값을 0으로 할 수 있다. 그리고 G_{sh} 또한 최대로 변할 수 있는 범위는 최대 전력지점을 벗어나지 않는 범위가 되어야 하므로

$$0 \leq G_{sh} \leq \frac{I_{sc} - I_{mpp}}{V_{mpp}} \quad (3.18)$$

식 (4.18)을 얻을 수 있다. 이를 태양광 패널의 I-V 특성곡선을 이용한 그림으로 표현하면 그림 3.5와 같이 나타낼 수 있다. 마지막으로 다이오드 이상계수 A 의 경우에는 재질에 따라서 값이 다를 수 있지만, 일반적으로 실리콘 타입의 결정형 패널이 많이 사용된다는 점을 고려하면 다음과 같다.

$$0 \leq A \leq 2 \quad (3.19)$$

이 값은 실리콘을 재료로 하는 경우에만 성립하는 값이다. 태양광 패널의 재료가 달라진다면 이 값 또한 수정되어야 한다.

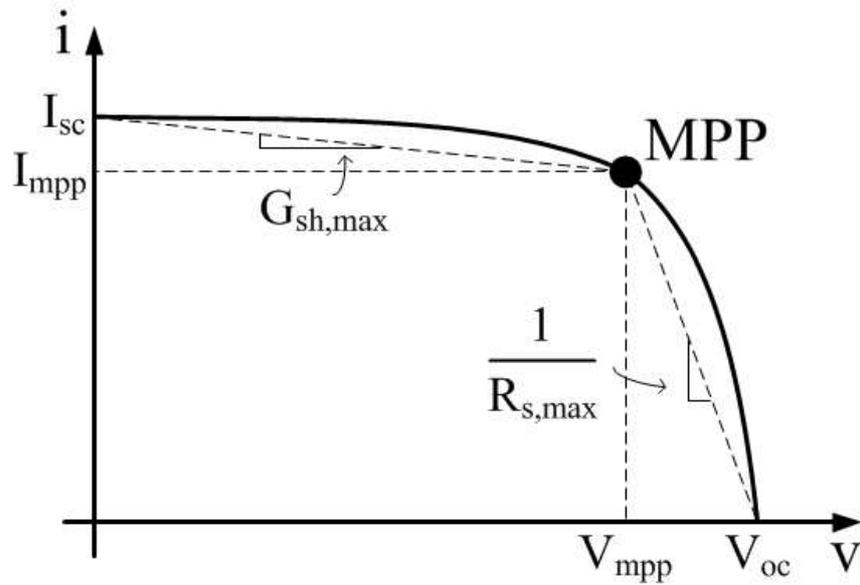


그림 3.5 탐색범위의 $R_{s,max}$ 와 $G_{sh,max}$ 의 정의

Fig. 3.5 Graphical method for $R_{s,max}$ and $G_{sh,max}$ of search range

그리고 파라미터를 추출하기 위한 목표함수는 매우 비선형적이므로 최적화의 실행과정에서 해가 발산하는 문제가 발생할 수 있다. 본 논문은 이 문제에 대해서 적절한 초기값을 제안함으로써 발산문제를 극복하고자 하였고 초기값의 선정조건으로는 앞서 제시한 탐색범위를 이용하였다. 각각의 해에 대한 탐색범위를 이용해서 제안방법의 초기값은 아래와 같이 제안된다.

$$X_1 = [R_{s,1} \ G_{sh,1} \ A_1] = [0, 0, 1] \quad (3.20)$$

4. 제안방법의 검증 및 응용

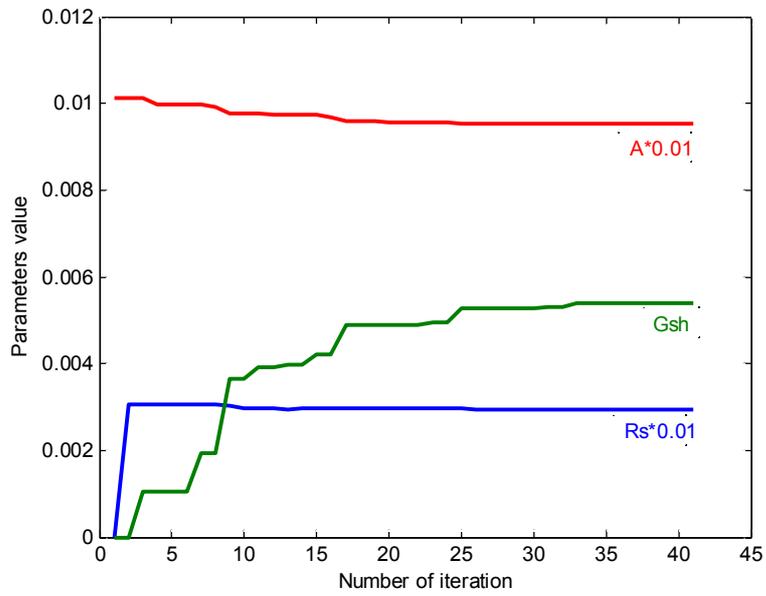
4장에서는 제안방법의 성능을 검증하고 제안방법의 활용방안에 대해서 설명한다. 제안모델의 성능은 기존의 모델과의 비교를 통해서 그 타당성을 증명하며, 활용방안의 경우 소프트웨어 시뮬레이션을 위한 소프트웨어 모델과 실제 태양광 시스템 실험을 위한 하드웨어 응용에 대해서 제안한다.

4.1 제안방법의 검증

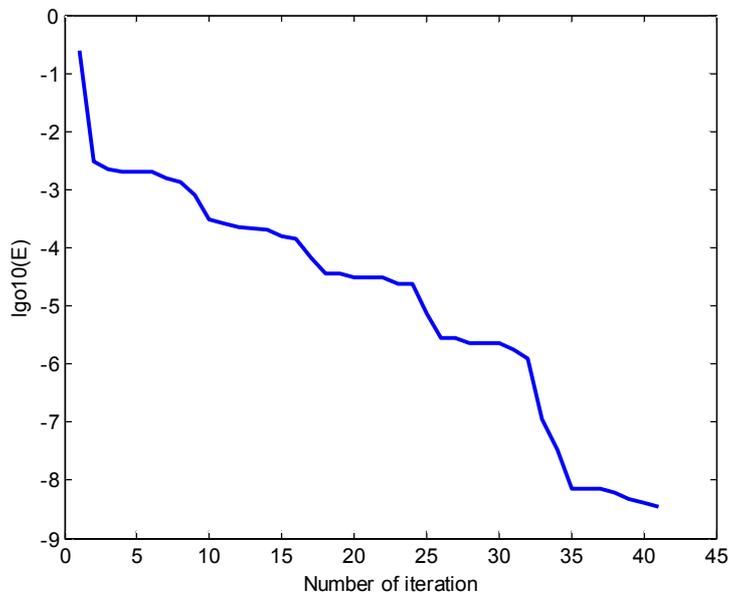
4.1.1 MATLAB과 PSIM을 이용한 모델링

제안방법의 소프트웨어적인 부분은 MATLAB 프로그램을 이용해서 구현이 가능하다. MATLAB은 사용자가 쉽게 프로그램을 작성하고 시뮬레이션을 할 수 있는 환경을 제공하는데 이를 이용하면 쉽게 제안방법이 사용하는 파워 최적화 방법을 비롯한 파라미터 추출 방법을 쉽게 구현 할 수 있다. 이때, 알고리즘의 어떻게 동작을 하는지 확인을 하기 위해서 그림 4.1과 같이 R_s , G_{sh} , A 의 각각의 파라미터와 목표함수 E 가 수렴되는 과정을 관찰하였다. 이때, 각 파라미터의 값은 실제 패널인 KC200GT의 데이터 시트 이용해서 추출한 것이다. 그림을 살펴보면, 먼저 그림 4.1의 (a)에서 싱글 다이오드 회로를 구성하는 파라미터 중 최적화를 통해서 구해지는 R_s , G_{sh} , A 의 값이 초기값으로부터 증가 또는 감소하면서 매 사이클마다 변하다가 결국에는 특정 값으로 수렴을 하는 것을 확인할 수 있다. 동시에 목표함수 E 의 값은 점점 감소하는 방향으로 알고리즘이 진행을 하는 것을 (b)의 그림을 통해서 확인 할 수 있고 이때 필요한 사이클은 41회가 소모됨을 확인할 수 있다.

이 과정을 통해서 구해진 각 파라미터 R_s , G_{sh} , A 를 이용해서 남은 두 개의 파라미터인 I_{ph} 와 I_0 또한 구할 수 있다. 최종적으로 5개의 모든 파라미터가 구해지게 되고 이 값을 PSIM의 회로모델에 적용하면 실제 태양광 패널과 관련된 I-V 혹은 P-V 특성곡선을 추출할 수 있게 된다. 이때, 사용되는 PSIM의 회로는 그림 4.2와 같이 구성된다. 이때, 태양광 패널의 등가모델이 결국에는 태양광 시스템과 연동하여 시뮬레이션을 실행할 것을 고려하면 반도체 소자의 열특성이나 스위칭 손실과 같은 회로적인 특성을 잘 반영 할 수 있는 PSIM의 모델이 더 유리하다고 판단하여 PSIM을 사용한 등가회로를 꾸몄다. 앞서 설명한 기존의 방법 또한 알고리즘은 MATLAB을 이용해서 구현하고 I-V 및 P-V 특성곡선은 추출한 파라미터를 PSIM으로 입력하여 그 결과를 추출하였다[18][19].



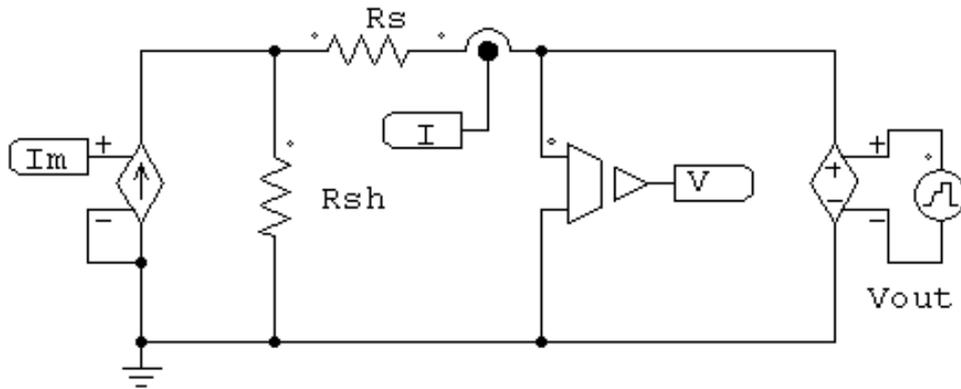
(a) Parameter convergence



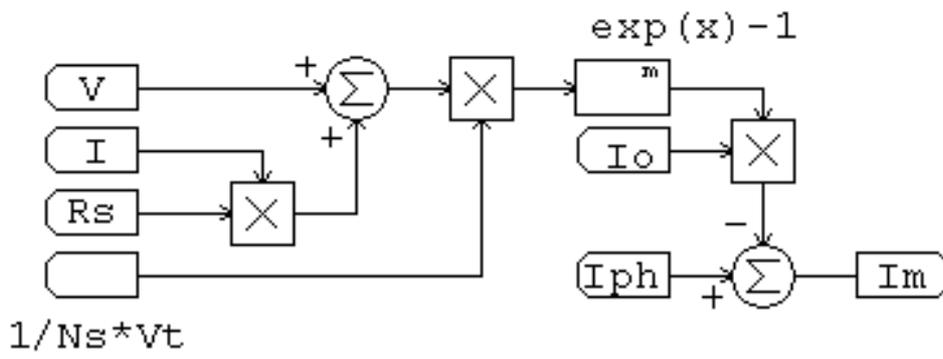
(b) Objective function evaluation

그림 4.1 제안방법의 최적화의 진행과정

Fig. 4.1 Optimization process of the proposed method



(a) Equivalent circuit



(b) Parameter calculation

그림 4.2 특성곡선 추출을 위한 PSIM 회로

Fig. 4.2 PSIM implementation to extract the characteristic curve

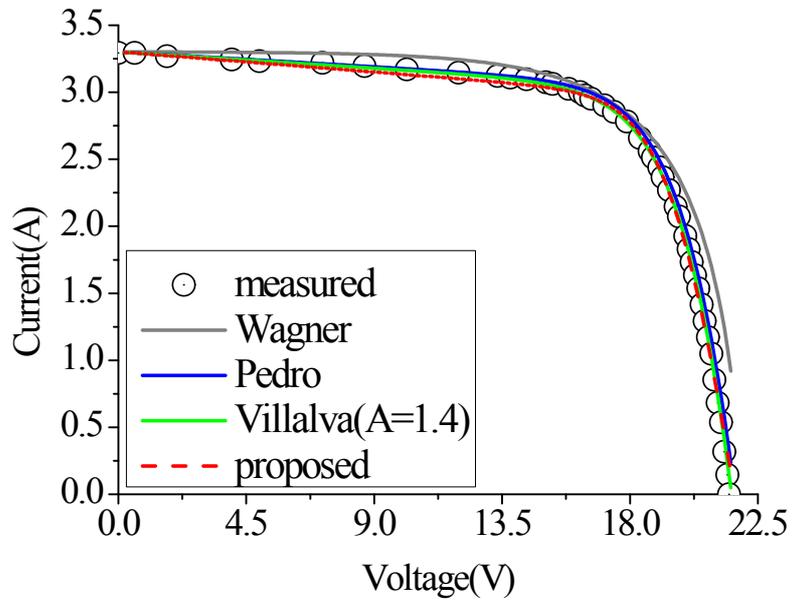
4.1.2 EN50530 규격

구현한 프로그램이 실제 태양광 패널의 출력특성을 얼마나 모사할 수 있는지 확인하기 위해서 9개의 다른 제조사의 실제 태양광 패널 - THERM Solartechnik AT50, BP Solar MSX60, Kyocera KC65GT, BP Solar MSX120, Shell Solar SQ160PC, Kyocera KC200GT, Samsung LPC241SM, Trina Solar TSM245PC, and Hanwha Solar SF260 - 을 선정하였다. 각각의 태양광 패널의 데이터 시트는 제조사에 따라서 제공하는 정보의 차이가 존재하지만 STC 조건 만큼은 모두 제공한다. 선정한 9개의 패널에 대한 정보는 아래의 표 4.1에 정리하였다. 이 중에서 3개의 패널 - AT50, MSX120, KC200GT - 에 대해서 I-V 및 P-V 특성 곡선을 3개의 기존방법과 제안방법을 이용해서 각각의 결과를 추출하였고 그 결과는 그림 4.3과 4.4의 결과와 같다. 그림 4.3과 4.4를 통해서 3개의 기존방법 및 제안방법의 파라미터 추출법이 실제 태양광 패널의 출력을 잘 모사할 수 있음을 확인할 수 있지만, 어떤 방법이 더 효율적이고 정확한 모델을 제공하는지는 알 수 없다. 그래서 본 논문에서는 유럽

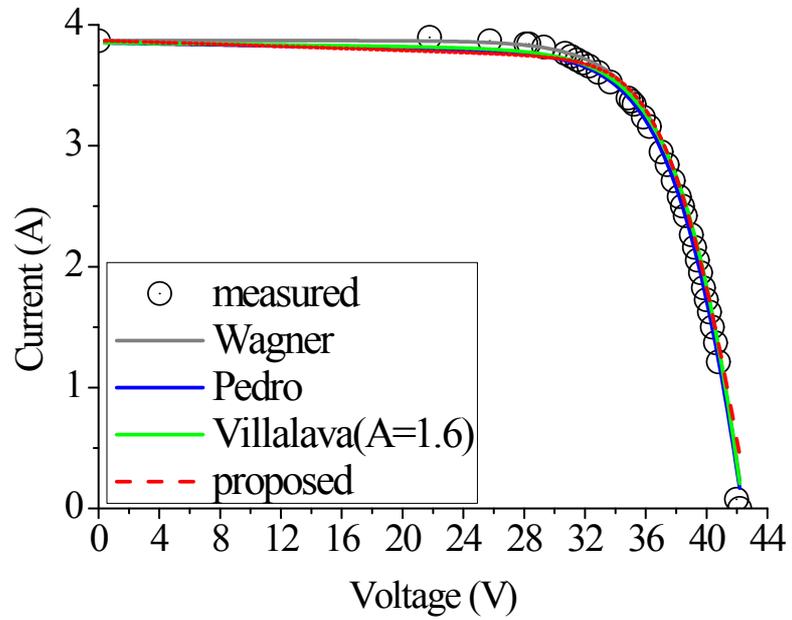
표 4.1 태양광 패널의 데이터 시트 정보

Table 4.1 Datasheet values of PV panels

	I_{sc} (A)	V_{oc} (V)	I_{mpp} (A)	V_{mpp} (V)	P_{mpp} (W)	N_s	k_i (A/°C)	k_v (mV/°C)
AT50	3.3	21.5	2.86	17.5	50	39	-	-
MSX60	3.8	21.1	3.5	17.1	60	36	$2.47 \cdot 10^{-3}$	-80
KC65GT	3.99	21.7	3.75	17.4	65	36	$1.59 \cdot 10^{-3}$	-82.1
MSX120	3.87	42.1	3.52	33.7	120	72	$2.47 \cdot 10^{-3}$	-80
SQ160PC	4.9	43.5	4.58	35	160	72	$1.4 \cdot 10^{-3}$	-161
KC200GT	8.21	32.9	7.66	26.7	200	54	$3.18 \cdot 10^{-3}$	-123
LPC241	8.54	37.4	8.01	30.1	241	60	$2.135 \cdot 10^{-3}$	-127.5
TSM245	8.68	37.5	8.13	30.2	245	60	$4.0796 \cdot 10^{-3}$	-120
SF260	8.4	44.3	7.76	36.1	280	72	$3.104 \cdot 10^{-3}$	-115.52



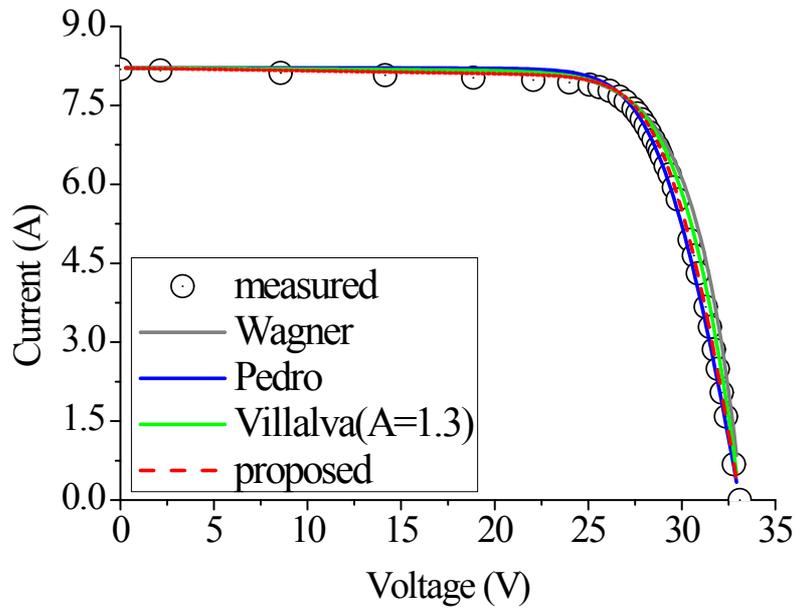
(a) AT50



(b) MSX120

그림 4.3 I-V 특성곡선

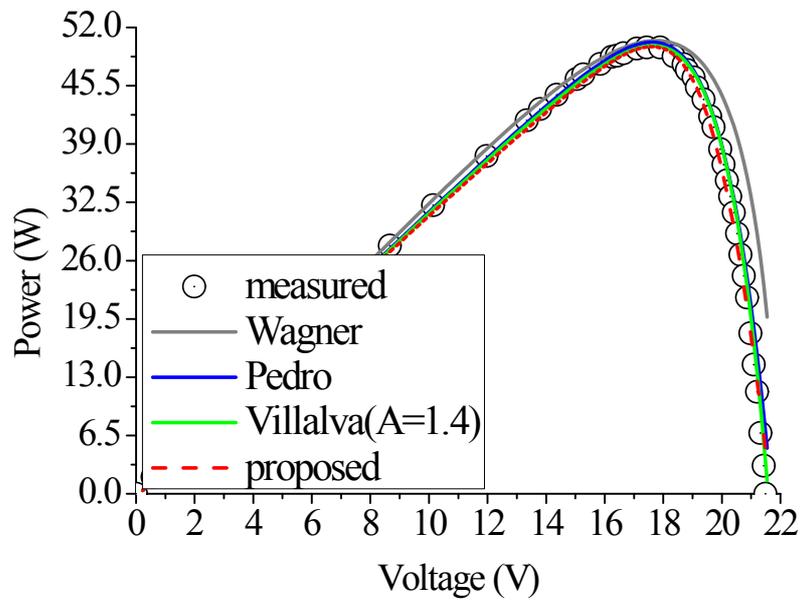
Fig. 4.3 I-V characteristic curve



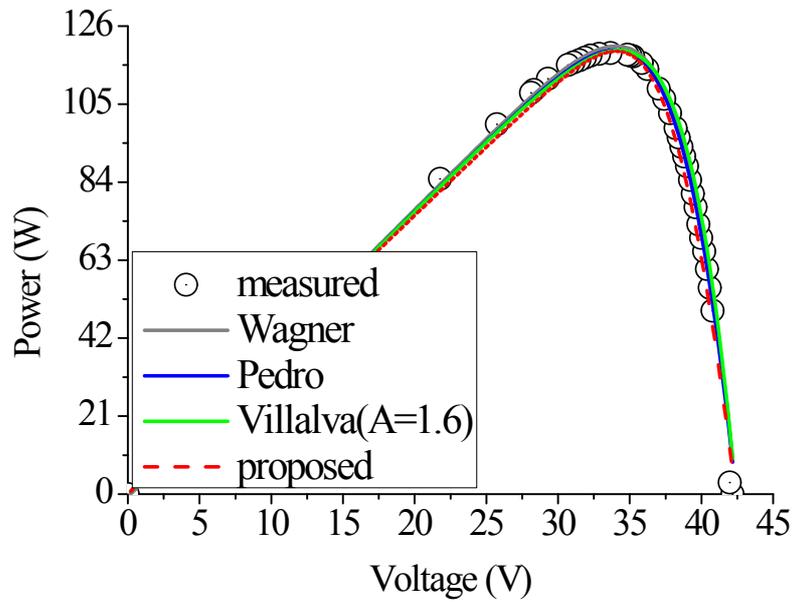
(c) KC200GT

그림 4.3 I-V 특성곡선 (계속)

Fig. 4.3 I-V characteristic curve (continued)



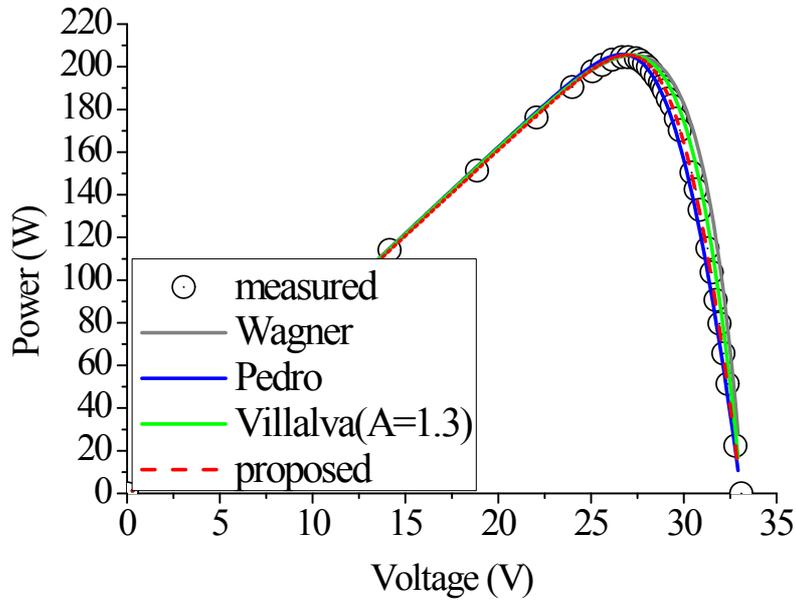
(a) AT50



(b) MSX120

그림 4.4 P-V 특성곡선

Fig. 4.4 P-V characteristic curve



(c) KC200GT

그림 4.4 P-V 특성곡선 (계속)

Fig. 4.4 P-V characteristic curve (continued)

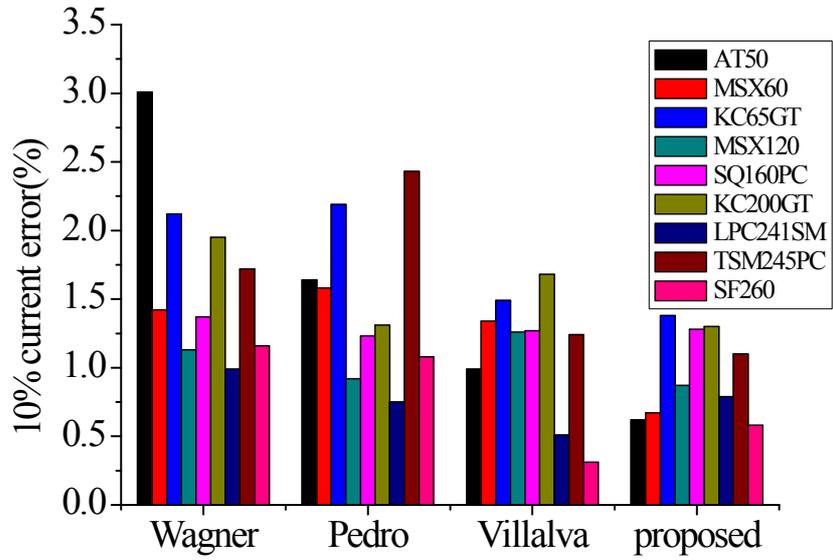
규격인 EN50530의 규격을 통해서 평가기준을 결정하였다.[20]

EN50530의 평가기준은 태양광 시뮬레이터의 MPP 부근에서의 허용오차에 대해서 규정하고 있으며, 이는 본 논문의 목적과 정확하게 맞아 떨어진다. 그래서 정의된 허용오차를 수식으로 표현하면 다음과 같다.

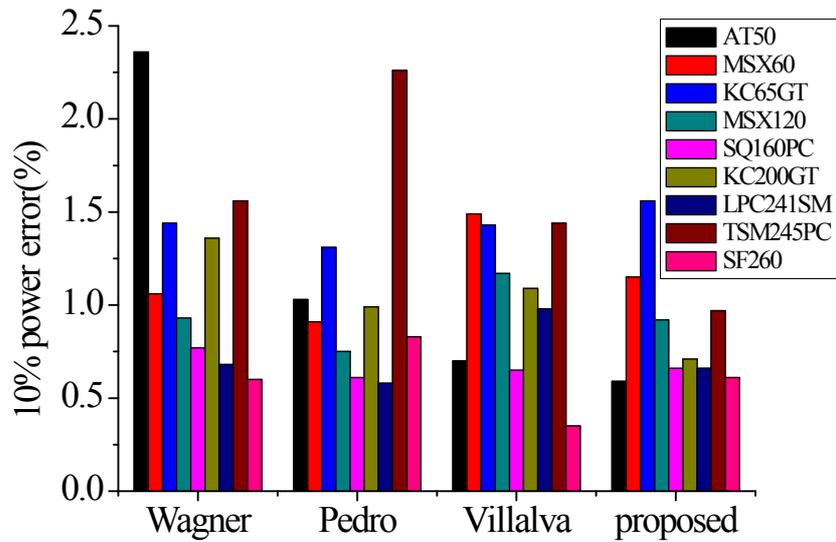
$$\epsilon_I(\%) = \frac{1}{0.2 V_{mpp}} \int_{V_{mpp} \pm 10\%} \left| \frac{i_s(v) - i_m(v)}{i_m(v)} \right| dv \cdot 100 \quad (4.1)$$

$$\epsilon_V(\%) = \frac{1}{0.2 V_{mpp}} \int_{V_{mpp} \pm 10\%} \left| \frac{p_s(v) - p_m(v)}{p_m(v)} \right| dv \cdot 100 \quad (4.2)$$

여기서 ‘m’은 실제 태양광 패널의 측정값을 의미하며, ‘s’는 시뮬레이션을 통해서 추출한 값이다. 결과는 앞서 선정된 9개의 실제 태양광 패널에 대해서 각각의 특성곡선을 추출하



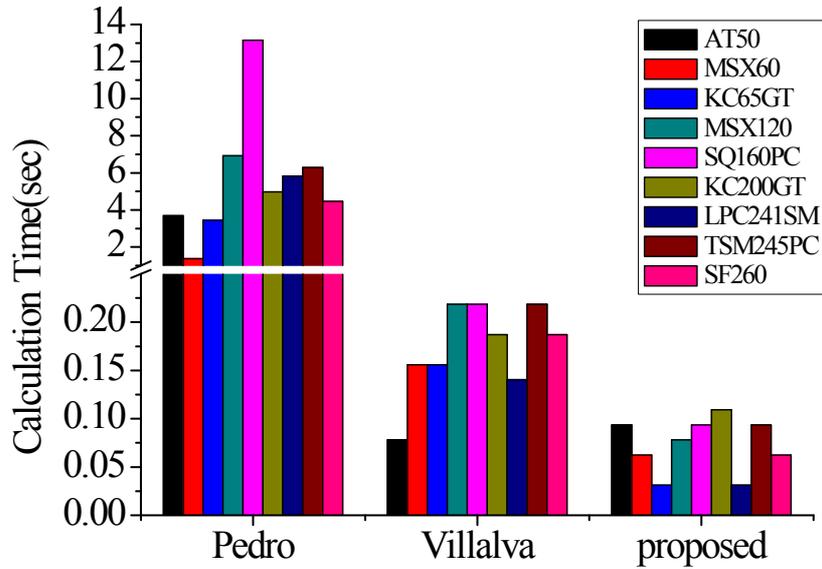
(a) 10% current error



(b) 10% voltage error

그림 4.5 알고리즘의 성능비교

Fig. 4.5 Comparison of algorithm performance



(c) Extraction time

그림 4.5 알고리즘의 성능비교 (계속)

Fig. 4.5 Comparison of algorithm performance (continued)

여 정리하였고 그 결과는 바-차트로 정리되어 그림 4.5와 같이 나타난다.

4.1.3 결과비교

우선, 그림 4.5의 바-차트에서 (a)와 (b)를 살펴보면, 제안방법의 결과가 좋은 경우가 많지만 좋지 않은 경우도 존재한다. 즉, 전체적인 성능은 제안방법이 양호함을 확인할 수 있지만 더 명확한 비교가 필요하다. 그리고 계산시간의 결과를 나타내는 (c)의 그림에서는 제안방법이 우수함을 확인할 수 있다.

실제로 대부분의 패널의 오차결과에서 제안방법의 결과가 양호함을 보이지만 더 정확한 비교를 위해서 본 논문은 통계적인 방법을 사용하였다. 이때, 사용한 통계적 방법은 평균값 (*Avg.*)과 표준편차 (*σ*)이다. 먼저, 평균의 경우 각 방법의 정확도를 비교하기 위해서 사용하였고 표준편차는 각 추출방법의 균등함을 비교하기 위해서 사용하였다. 여기서 표준편차는 평균의 값에 영향을 받기 때문에 더 정확한 균등성의 비교는 표준편차를 평균으로

표 4.2 각 방법의 통계적 비교검증

Table 4.2 Statistical verification for each algorithm

	Current error (%)			Voltage error (%)			Calculation time (sec.)		
	<i>Avg.</i>	σ	$\sigma / Avg.$	<i>Avg.</i>	σ	$\sigma / Avg.$	<i>Avg.</i>	σ	$\sigma / Avg.$
Wagner	1.652	0.637	0.385	1.196	0.555	0.464	-	-	-
Pedro	1.459	0.563	0.386	1.03	0.513	0.498	3.3	5.571	0.592
Villalva	1.121	0.447	0.399	1.033	0.4	0.387	0.046	0.173	0.268
Proposed	0.954	0.315	0.33	0.87	0.321	0.37	0.028	0.073	0.386

나누면 얻을 수 있는 변동계수 ($\sigma / Avg.$)를 사용하였다. 그리고 그 결과는 표 4.2에 나타났다.

표 4.2의 결과를 분석해보면, 평균값의 경우 전류와 전압의 허용오차에서 제안방법이 가장 좋은 것을 확인할 수 있다. 그때, 허용오차의 평균은 1%를 넘지 않는 것을 확인할 수 있으며, 이는 EN50530의 규격을 만족하는 결과이다. 다른 기존의 방법 중에서 Pedro와 Villalva 방법의 전압오차는 1%가 조금 넘는 값으로 EN50530의 규격에 근접하지만 전류오차가 큰 것을 확인할 수 있다. 다음으로 표준편차와 평균을 이용해서 구한 변동계수를 살펴보면, 제안방법의 변동계수가 가장 작은 것을 확인할 수 있다. 이것이 의미하는 것은 제안방법이 패널의 출력레벨이나 종류에 상관없이 항상 균일한 결과를 제공함을 의미한다. 결과적으로 제안방법의 결과가 우수함을 확인할 수 있다. 하지만 실제 태양광 시스템의 시뮬레이션에 제안방법을 적용하기 위해서는 하나의 소프트웨어로 구현이 되며, 실제 태양광 시스템의 구동환경처럼 변하는 온도와 일사량에 따라 변하는 출력특성을 제공할 수 있는 모델이 필요하다.

4.2 제안방법의 소프트웨어 응용

제안모델이 실제 태양광 시스템의 시뮬레이션에 사용이 되기 위해서는 우선은 변하는 온도와 일사량에 따른 태양광 패널의 출력특성을 표현할 수 있어야 하며, 다음으로 하나의 시뮬레이션 프로그램을 이용해서 구현이 되어 시스템 시뮬레이션과 연동이 가능하여야 한다. 이를 위해서 본 논문은 제안방법의 소프트웨어 응용방법에 대한 연구도 병행하였다.

4.2.1 제안모델의 가변환경 모델링

앞서 설명한 기존방법에서, 태양광 패널의 온도 및 일사량에 대한 가변환경 모델링은 변하는 온도 및 일사량에 대한 모델을 구현하기 위해서 싱글-다이오드 모델의 내부 파라미터인 R_s , G_{sh} , A 의 값을 고정으로 두고 I_{ph} , I_0 와 같은 전류 값이나 V_{oc} 의 전압만을 변화시키는 것으로 모델링을 하였다. 이 방법은 데이터시트에서 제공하는 단락전류 I_{sc} 와 개방전압 V_{oc} 의 온도상수인 k_i 와 k_v 를 이용하면 쉽게 모델링을 할 수 있기 때문에 많이 사용되는 방법이다.

하지만, 보다 정확한 모델을 구현하기 위해서는 내부 파라미터의 변화 또한 반영을 해주는 것이 좋다.[21][23] 그러나 태양광 패널의 제조사에서 제공하는 정보는 I_{mpp} 나 V_{mpp} 의 온도상수를 제공하지 않고 있기 때문에 모델링이 어렵다. 이 문제를 극복하기 위해서 제안방법은 아래의 근사식을 이용해서 I_{mpp} 와 V_{mpp} 의 온도상수를 정하였다.

$$k_{i,mpp} \cong k_i, k_{v,mpp} \cong k_v \quad (4.3)$$

이 수식은 문헌[21][22]를 근거로 하는 근사식으로서, 내부 파라미터를 보정해주기 때문에 기존의 모델링 방법에 비해서 정확한 모델을 얻을 수 있다. 그리고 태양광 패널의 전류는 일사량 S 에 정비례하므로 식 (4.4)를 정의할 수 있고

$$I_{sc} = I_{sc,STC} \frac{S}{S_{STC}} \quad (4.4)$$

전압의 경우에는 일사량에 대해서 지수함수로 비례하므로

$$V_{oc} = V_{oc,STC} + N_s A_{STC} V_T \ln \left(\frac{S}{S_{STC}} \right) \quad (4.5)$$

의 식을 얻을 수 있다[23][24][25]. 결국, 식 (4.3)의 근사식을 이용해서 얻을 수 있는 제안 방법의 가변환경에 대한 수식은 아래와 같다.

$$I_{sc} = I_{sc,STC} \frac{S}{S_{STC}} [1 + k_i (T - T_{STC})] \quad (4.6)$$

$$V_{oc} = V_{oc,STC} + N_s A_{STC} V_T \ln \left(\frac{S}{S_{STC}} \right) + k_v (T - T_{STC}) \quad (4.7)$$

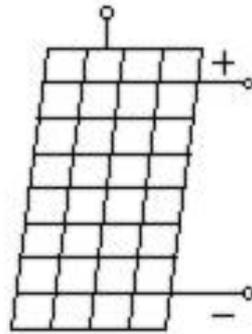
$$I_{mpp} = I_{mpp,STC} \frac{S}{S_{STC}} [1 + k_{i,mpp} (T - T_{STC})] \quad (4.8)$$

$$V_{mpp} = V_{mpp,STC} + N_s A_{STC} V_T \ln \left(\frac{S}{S_{STC}} \right) + k_{v,mpp} (T - T_{STC}) \quad (4.9)$$

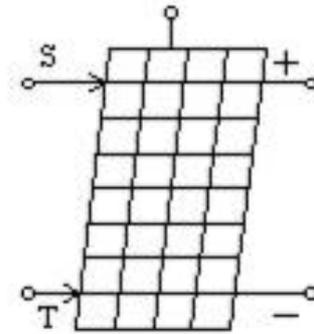
수식에서 보이듯이 가변환경 데이터를 사용하기 위해서는 기본적으로 STC에서의 조건을 필요로 한다. 그래서 가변환경을 사용하기 전에 STC에서의 데이터를 먼저 추출해 줄 필요가 있다. 하지만 위의 과정을 앞선 4.1.1절에서와 같이 MATLAB과 PSIM을 이용해서 따로 구현을 하게되면 파라미터 추출과정의 자동화가 어렵다. 결국 하나의 프로그램을 통해서 제안모형을 구현할 필요가 있다. 제안모형의 사용성의 확대를 위해서 본 논문은 PSIM의 C block을 이용하였고 PSIM의 경우 태양광 PCS와의 연계 시뮬레이션에 유리하다고 판단하여 PSIM을 통합을 위한 툴로 선정하였다.[26]

4.2.2 PSIM의 태양광모형

제안모형을 구현하기에 앞서 C block으로 구현된 제안모형의 타당성을 검증하기 위해서 기존의 PSIM에서 제공하는 태양광모형의 특징부터 분석하였다. PSIM은 Solar모듈 라이브러리 안에 자체적으로 2개의 태양광 모형을 제공하고 있는데 함수모형과 물리적 모형이다. 각 모형은 PSIM에서 그림 4.6과 같이 확인할 수 있다. 여기서 함수모형은 표 4.3과 같이 태양광 패널의 데이터시트에서 I_{sc} , V_{oc} , I_{mpp} , V_{mpp} 의 값만 알고 있으면 적은 정보량으로도 쉽게 모델링을 할 수 있다는 장점을 가지지만, 변하는 온도 및 일사량에 대한 출력특성은 표현이 불가능해서 실제 태양광 시스템의 시뮬레이션에 사용하는 것이 어렵다.[19]



(a) functional model



(b) physical model

그림 4.6 PSIM의 태양광 패널 모델

Fig. 4.6 PV panel model in PSIM

표 4.3 함수모델의 입력 파라미터

Table 4.3 Input parameters for functional model

User setting parameter	Value	Unit
Open circuit voltage(Voc)	80	V
Short circuit current(Isc)	5.9	A
Maximum power voltage(Vmpp)	62	V
Maximum power current(Imp)	4.8	A

반면에 물리모델의 경우 함수모델보다 더 많은 정보를 입력해야 사용이 가능하지만, 변하는 온도 및 일사량에 대한 출력특성의 변화를 표현할 수 있기 때문에 실제 태양광 시스템의 시뮬레이션에서 많이 사용되고 있다. 따라서 본 논문에서는 이하 물리모델을 PSIM 모델로 지칭하기로 한다. PSIM 모델은 기본적으로 싱글-다이오드 회로모델로서 태양광 패널의 물리적 특성을 표현하며, PSIM 모델을 사용하기 위한 입력 파라미터는 표 4.4와 같이 나타난다. 그리고 PSIM모델의 기본적인 수식은 다음과 같다.

$$i = i_{ph} - i_d - i_r \quad (4.10)$$

$$i_{ph} = I_{s0} \frac{S}{S_{STC}} + k_i (T - T_{STC}) \quad (4.11)$$

$$i_d(T) = I_o \left(e^{\frac{q v_d}{A k T}} - 1 \right) \quad (4.12)$$

$$I_o(T) = I_{s0} \left(\frac{T}{T_{STC}} \right)^3 e^{\frac{q E_g}{A k} \left(\frac{1}{T_{STC}} - \frac{1}{T} \right)} \quad (4.13)$$

표 4.4 물리모델의 입력 파라미터

Table 4.4 Input parameters for physical model

User setting parameter	Value	Unit
Number of cells(Ns)	36	-
Standard light intensity(S0)	1000	W/m ²
Ref. temperature(Tref)	25	K
Series Resistance(Rs)	0.008	Ω
Shunt Resistance(Rsh)	1000	Ω
Short circuit current(Isc0)	3.8	A
Saturation current(Is0)	2.16e-8	A
Band energy(Eg)	1.12	eV
Ideality factor(A)	1.2	-
Temperature coefficient(Ct)	0.0024	A/K
Coefficient(Ks)	0	-

$$i_r = \frac{v_d}{R_{sh}} \quad (4.14)$$

수식 (4.10)은 싱글-다이오드 회로를 표현한 식으로 i 는 광전류 i_{ph} 로부터 다이오드 전류 i_d 와 누설전류 i_r 을 뺀 값으로 표현된다. 이때, i_{ph} 는 식 (4.11)과 같이 I_{sc0} 의 값으로부터 구할 수 있으며, I_{sc0} 는 STC에서의 단락전류를 의미하고 STC를 기준으로 외부의 일사량 (S)과 온도 (T)에 따라서 변한다. i_d 는 암전류 (I_0)를 이용해서 얻을 수 있으며, 식 (4.13)에서 I_{s0} 는 STC에서의 암전류를 의미한다. 마지막으로 i_r 은 다이오드 전압 (v_d)를 이용해서 얻을 수 있다.

수식을 통해서 PSIM 모델을 사용하기 위해서는 사용자가 직접 R_s , G_{sh} , A 의 값을 결정해 줘야 한다. 그러나 이 값을 구하기 위해서는 추가적인 파라미터 추출법이 필요하며, 결국 PSIM 모델의 동작원리는 그림 4.7와 같은 개념도로 그려질 수 있다. 그림에서 알 수 있는 PSIM 모델의 특징은 다음과 같다. 먼저, 수식을 살펴보면 PSIM 모델은 내부 파라미터 R_s , R_{sh} , A 의 변화는 고려하지 않으며, I_{ph} 와 I_0 의 수식을 변화시켜서 가변온도 및 일사량에 대한 정보를 제공하는 것을 확인할 수 있다. 이때, I_{ph} 와 I_0 의 STC에서의 값과 싱글-다이오드 회로의 파라미터 R_s , R_{sh} , A 의 값을 구하기 위해서 PSIM 모델은 추가적인 파라미터 추출 방법이 필요하며, 이 과정을 위해서 PSIM은 Solar Module Utility라는 별도의 파라미터 추출 Tool을 제공하고 있다. 그러나 이 프로그램을 이용해서 파라미터를 추출하는 과정에서 시

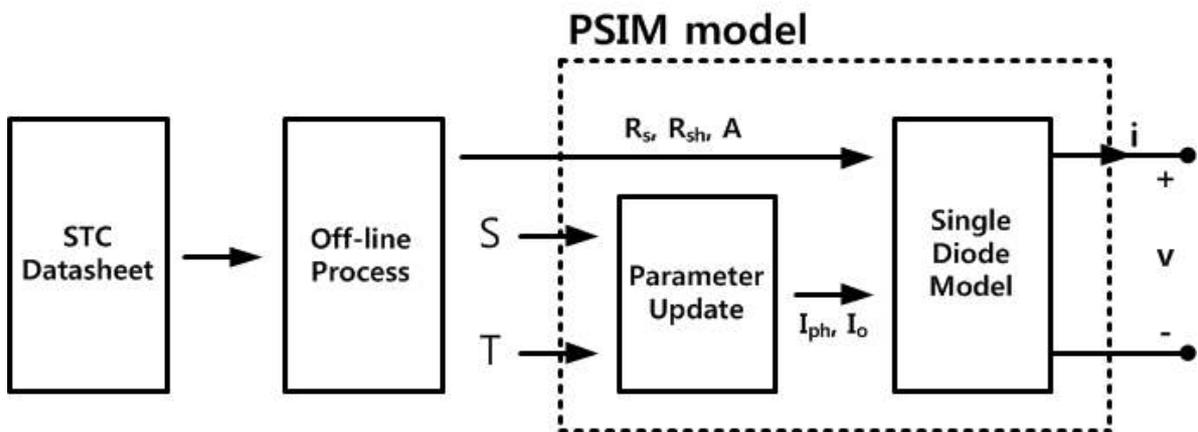


그림 4.7 PSIM 모델의 동작원리

Fig. 4.7 Performance of the PSIM model

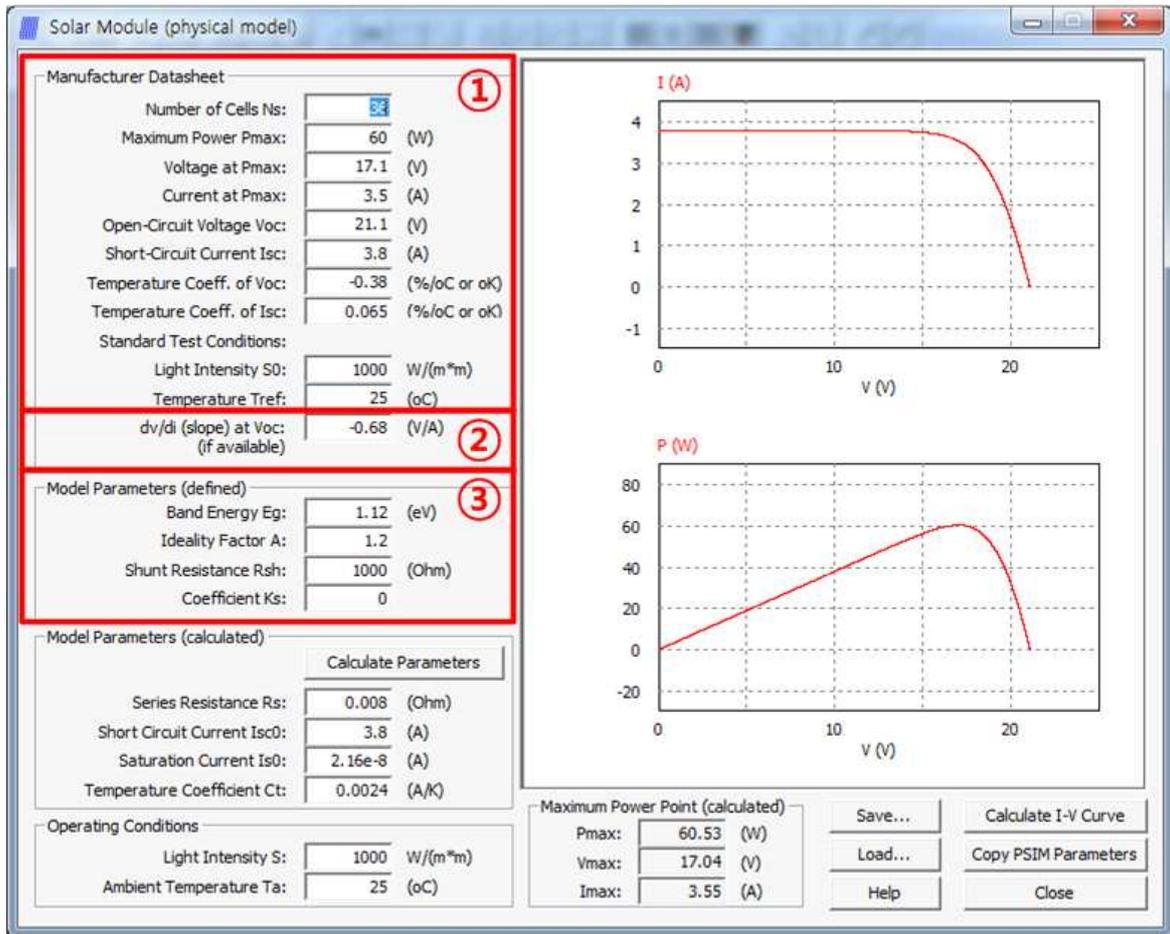


그림 4.8 PSIM의 태양광 모듈 프로그램

Fig. 4.8 Solar Module Utility of the PSIM

행착오적인 방법이 필요한데 이는 모델의 오차에 영향을 줄 수 있다.

프로그램의 작동방법은 아래와 같다. 먼저 사용자 인터페이스는 그림 4.8과 같으며[27], 관련수식은 식 (4.14)에서 (4.17)에 정의하였다.[28][29] 여기서 사용자가 필요로 하는 데이터를 입력한 뒤에 Calculation Parameters 버튼을 누르면 PSIM의 모델을 구성하는데 필요한 파라미터 R_s , I_{sc0} , I_{s0} 를 얻을 수 있다. 파라미터의 추출을 위해서 사용자는 먼저 Manufacturer Datasheet (①)의 부분에 데이터시트로부터 얻을 수 있는 정보들을 입력하고 STC에서의 온도 및 일사량 조건을 입력해야 한다. 이때, 핵심은 식 (4.17)을 이용해서 R_s 를 구하는 것으로 식을 사용하기 위해서 slope at Voc (②)의 부분에 V_{oc} 에서의 기울기를 데이터시트 값과 함께 입력해야 한다. 기울기는 데이터시트에서 제공하는 I-V 특성곡선으로부터

터 작도를 통해서 얻을 수 있으며, Voc에서의 기울기를 입력하면 식(4.17)에서 Voc에서의 기울기를 결정할 수 있다. 그리고 X_V 를 구하기 위해서는 식 (4.15)에서 볼 수 있듯이 사용자가 A와 Rsh의 값을 추가로 입력해야 하며, 식 (4.15)에서 필요한 I_o 는 식 (4.16)을 통해서 얻을 수 있다. 이 과정은 그림 4.8의 Model Parameters (defined) (③)부분에 사용자가 직접 Band Energy (E_g), Diode Factor(A), Shunt Resistance (Rsh)를 입력하도록 구성되며, 최종적으로 Manufacturer Datasheet(①, ②)에서 입력한 값과 함께 식 (4.17)에 필요한 모든 파라미터를 알 수 있으므로 R_s 를 구할 수 있다.

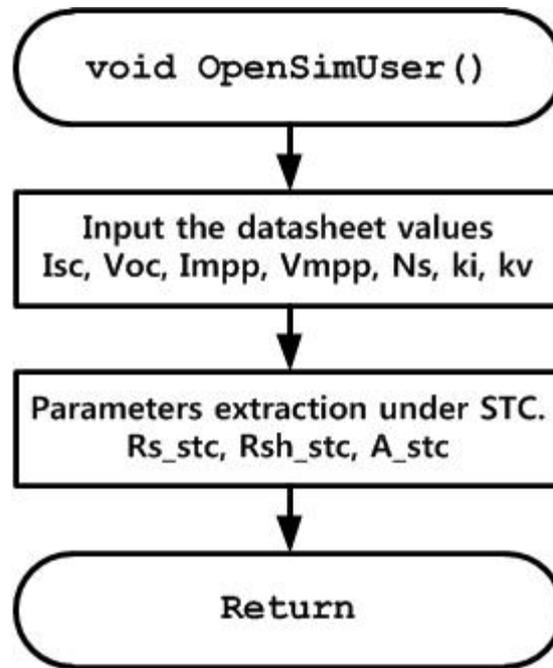
$$\left. \frac{di}{dv} \right|_{@V_{oc}} = \frac{-X_v}{1 + R_s X_v} \quad (4.14)$$

$$X_V = \frac{q}{AkT} I_o e^{\frac{qV_{oc}}{AkT}} - \frac{1}{R_{sh}} \quad (4.15)$$

$$I_o = \frac{I_{sc}}{e^{\frac{qV_{oc}}{AkT}} - 1} \quad (4.16)$$

$$R_s = - \left. \frac{di}{dv} \right|_{@V_{oc}} - \frac{1}{X_V} \quad (4.17)$$

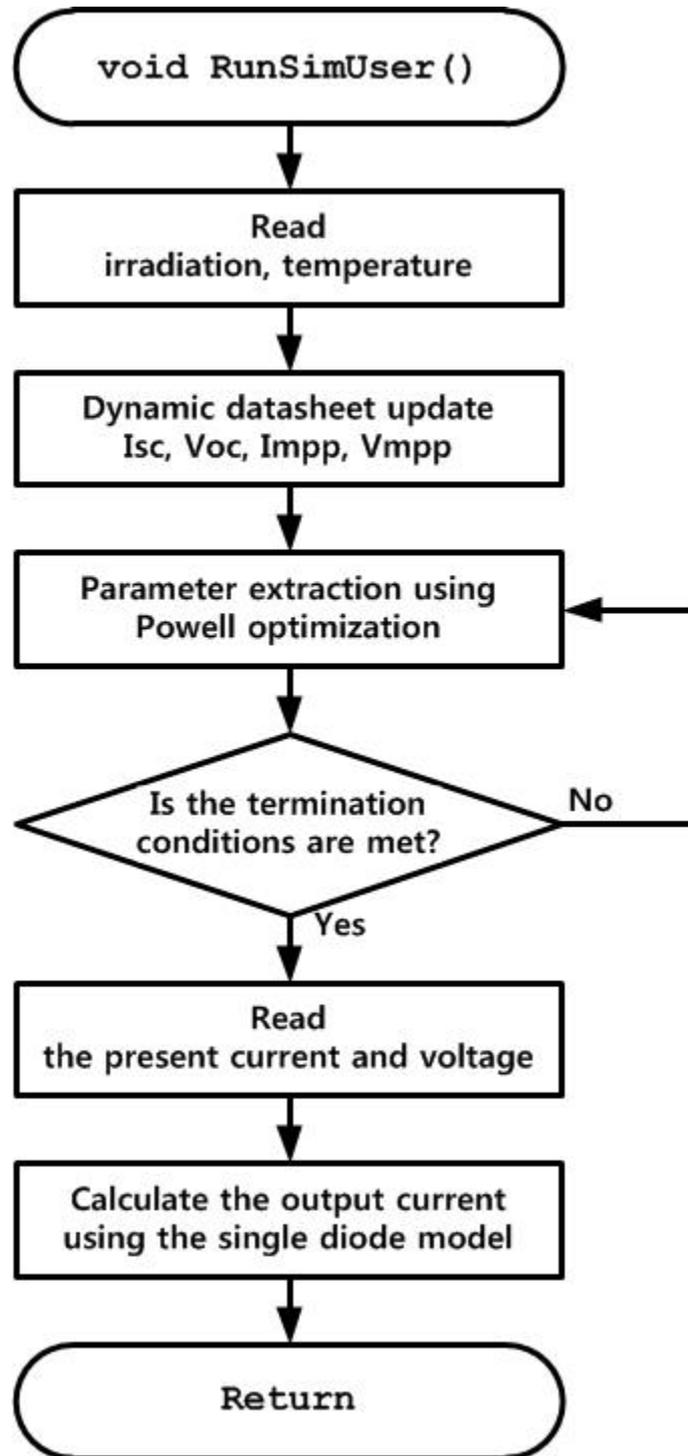
Solar Module을 이용해서 구한 파라미터는 표 4.4에 표시된 PSIM 모델의 파라미터로 사용이 되며, 표에서 N_s , S_0 , T_{ref} 는 그림 4.8의 ①로부터 결정되는 것을 확인할 수 있다. 그리고 I_{s0} (I_o)의 추출을 위해서 I_{sc} (I_{sc0})를 데이터시트로부터 입력하며, R_s 의 계산을 위해서 R_{sh} , E_g , A를 정의한다. 표에서 C_1 는 ①의 Temperature coeff. of I_{sc} 로부터 계산되는 값이며, K_s 는 일사량에 따른 온도변화에 대한 상수이지만, 쓰지않는 경우에는 0으로 둔다. 최종적으로 표 4.4의 값과 PSIM 모델의 식 (4.10)에서 (4.14)를 이용해서 완성된 싱글-다이오드 등가모델을 이용해서 가변온도와 일사량에서의 출력특성을 얻을 수 있다.



(a) OpenSimUser part

그림 4.9 제안모델의 C block 구현

Fig. 4.9 C block implementation of the proposed model



(a) RunSimUser part

그림 4.9 제안모델의 C block 구현 (계속)

Fig. 4.9 C block implementation of the proposed model (Continued)

4.2.3 C block을 이용한 제안모델의 구현

본 논문에서 제안하는 모델을 PSIM만으로 구현하기 위해서 C block을 이용할 수 있다. C block의 경우 추가적인 프로그램의 도움이 필요하지 않고 PSIM과 C 언어만으로도 알고리즘을 모두 표현할 수 있는 장점을 가진다. 이때, C block은 여러 부분으로 나뉘어 있는데 사용자가 주로 사용하게 되는 부분은 OpenSimUser와 RunSimUser 부분이다. 이를 통해서 구현된 제안모델의 장점은 기존 PSIM 모델과 다르게 싱글-다이오드 회로의 내부 파라미터의 변화도 고려하고 사용자의 개입을 최소화 하여 사용자의 파라미터 선택으로 인한 영향도 줄여서 태양광 모델의 정확도를 향상시킬 수 있다는 것이다.

제안모델의 구현을 위해서 C block의 각 부분에 대한 설명을 하면, 우선 OpenSimUser는 PSIM이 시뮬레이션의 시작과 동시에 한 번만 불러오는 부분으로 기본적으로 사용되는 글로벌 함수나 수식이 정의되는 부분이다. 본 논문에서는 태양광패널의 수식에 사용되는 함수 및 Powell 최적화 방법과 내부적으로 사용되는 황금분할법 (GoldenSection)이 OpenSimUser에 정의되었다. 다음으로 RunSimUser는 시뮬레이션이 실행되는 동안 매 스텝마다 호출이 되는 부분으로 실제로 반복문이 실행되는 부분이다. 여기서 OpenSimUser에 정의된 Powell과 GoldenSection을 반복적으로 호출하면서 최종적인 결과를 찾는 것이다. 그 결과 C block은 그림 4.9과 같은 흐름도로 표현될 수 있고 이를 이용한 제안방법의 전체적인 개념도는 그림 4.10과 같다. 그림에서 볼 수 있듯이 C block으로 구현된 제안방법은 의

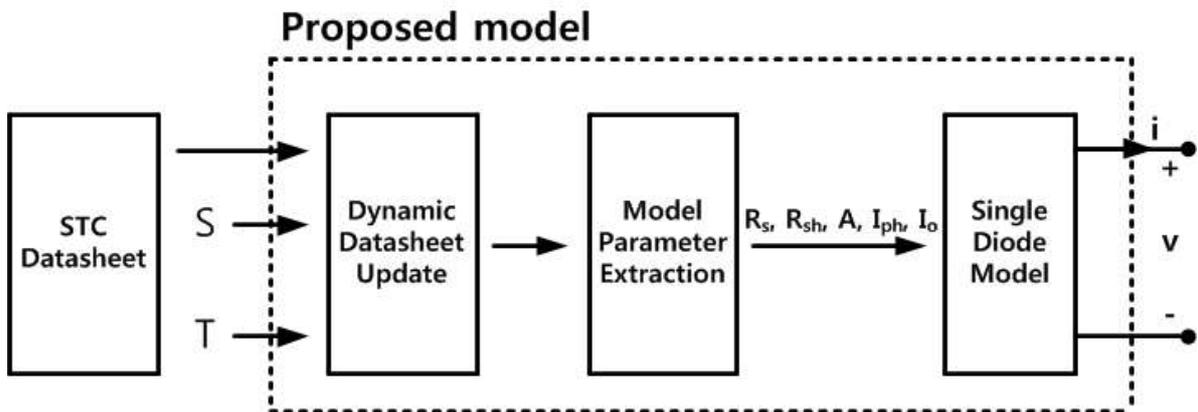


그림 4.10 제안모델의 동작원리

Fig. 4.10 Performance of the proposed model

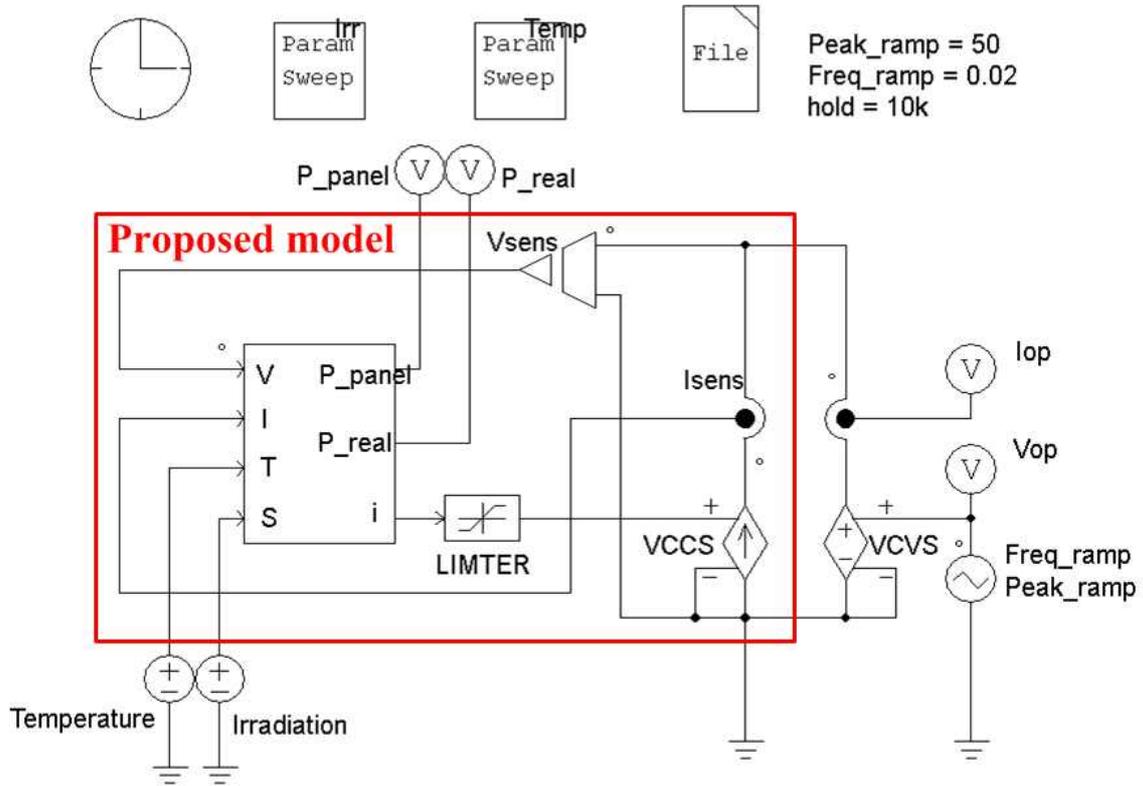


그림 4.11 제안모델의 소프트웨어 구현

Fig. 4.11 Software implementation of the proposed model

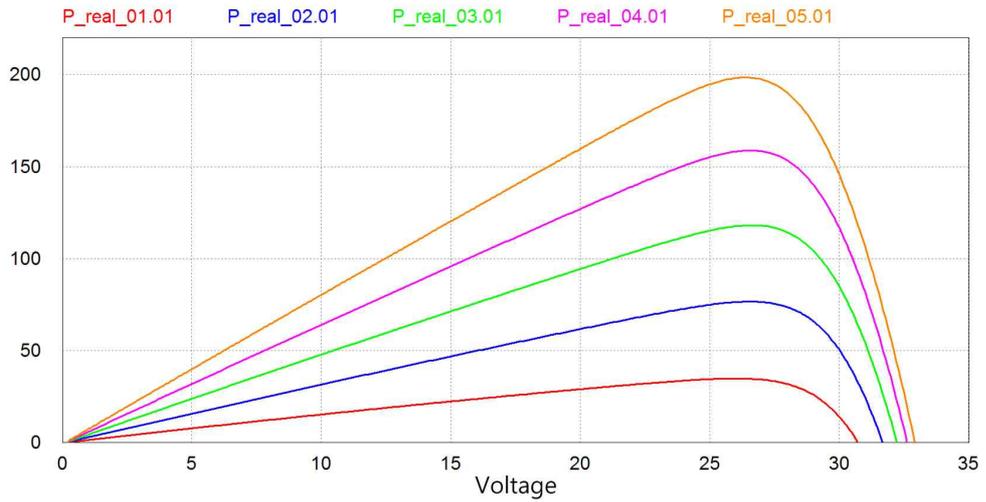


그림 4.12 제안모델의 출력특성 결과 (KC200GT)

Fig. 4.12 Characteristic results of the proposed model (KC200GT)

부에서 따로 파라미터 추출을 해줄 필요가 없을 뿐더러 사용자가 파라미터를 직접 선택할 필요도 없이 태양광 패널의 제조사가 보장하는 데이터시트의 값만으로 등가회로 구성에 필요한 파라미터를 추출할 수 있음을 알 수 있다. 이때, 제안모델은 동적 데이터시트 갱신을 통해서 정확도 또한 보장할 수 있다.

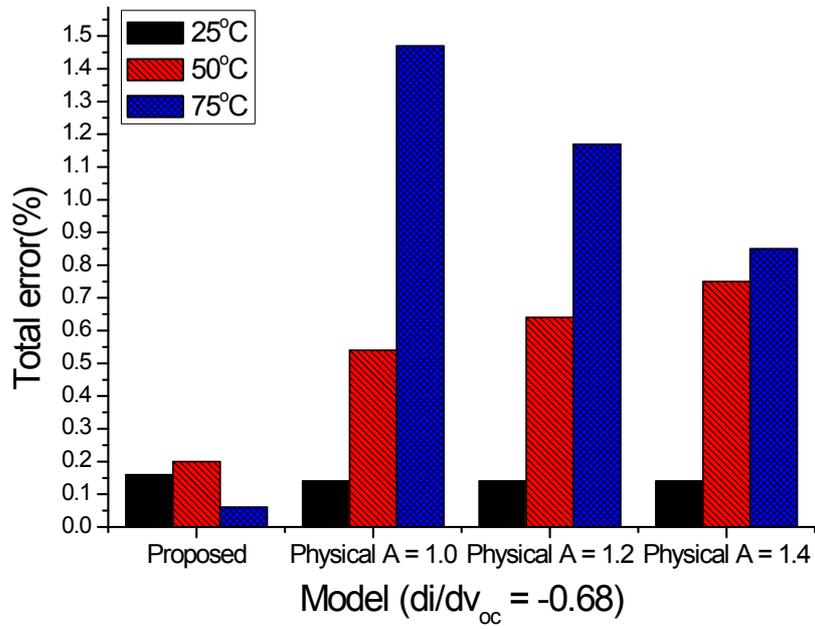
최종적으로 C block을 이용해서 회로모델로서 제안모델을 구성하면 그림 4.11과 같은 회로를 얻을 수 있다. 이때, 그림에서 VCCS (Voltage-Controlled Current Source)를 사용해서 최종적인 회로 모델을 구현 하였는데, 기본적인 태양광 패널의 출력은 전류의 형태이기 때문에 전압의 출력을 내는 C block의 출력을 전류의 형태로 변환시키기 위해서 사용되었다. 이를 이용하면 그림 4.12와 같이 가변온도 및 일사량에 대한 태양광 패널의 출력특성을 얻을 수 있다. 이렇게 구현된 제안모델은 기존 PSIM 모델과의 비교검증을 통해서 그 타당성을 확인할 것이다.

4.2.4 PSIM 모델과의 비교검증

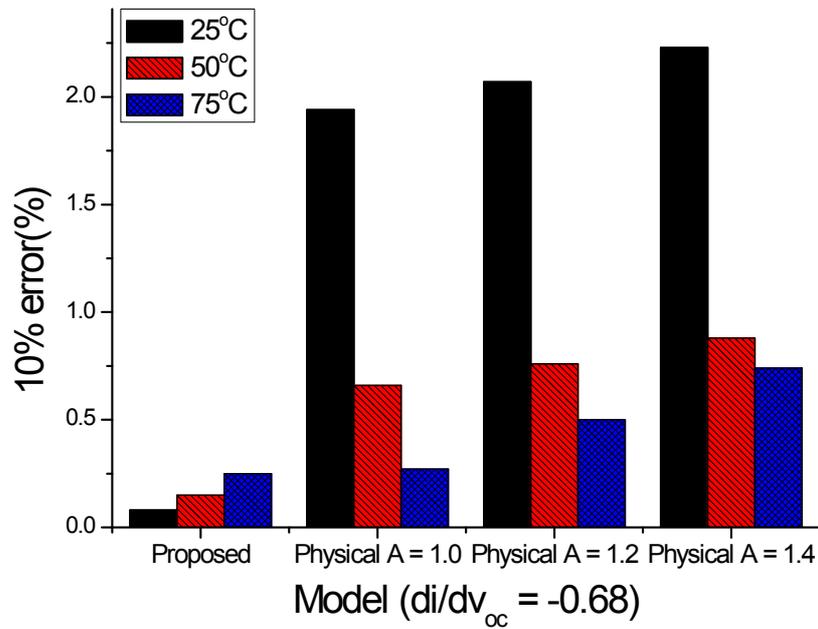
다음으로 PSIM으로 구현된 제안모델의 타당성을 검증하기 위해서 앞선 제안모델의 검증에서 사용했던 EN50530의 규격을 다시 사용했으며, 이때 실제 패널의 데이터시트 정보는 온도와 일사량의 정보가 명확하게 제공되는 KC200GT의 것을 사용하였다. 그리고 PSIM 모델은 사용자가 입력한 파라미터에 따라서 패널의 특성이 조금씩 바뀌므로 여러 경우에 대해서 모델을 추출하였다. 그 결과는 그림 4.13과 4.14에 나타냈다.

그림에서 PSIM 모델의 사용자 입력 파라미터 A와 V_{oc} 에서의 기울기는 인위적으로 변화되는 것을 확인할 수 있다. 이때, V_{oc} 의 기울기를 -0.68 로 고정하고 A를 1.0, 1.2, 1.4으로 각각 변화시킨 경우와 A는 1.2로 고정하고 V_{oc} 에서의 기울기는 각각 -0.58 , -0.68 , -0.78 의 값을 사용한 경우에 대해서 PSIM 모델의 출력특성을 얻었다. 그리고 온도와 일사량의 변화에 대해서 각각 그림 (a), (b), (c), (d)로 나누었다. (a)는 기울기를 고정한 후 일사량을 200에서 1000 W/m^2 까지 200 W/m^2 간격으로 변화시키면서 모델의 전체오차를 구한 것이고 (b)는 MPP의 $\pm 10\%$ 지점에서의 오차를 구한 것이다. 그리고 (c)와 (d)는 A를 고정한 후에 온도를 25에서 75°C 까지 25°C 의 간격으로 바꾸면서 모델의 전체오차와 10% 오차를 구한 것이다.

먼저, 그림 4.13의 온도변화에 대한 결과를 살펴보면, 제안모델의 결과가 우수함을 확인할 수 있다. 즉, 사용자가 PSIM 모델의 파라미터를 잘 선택하였을 경우에는 정확도가 더



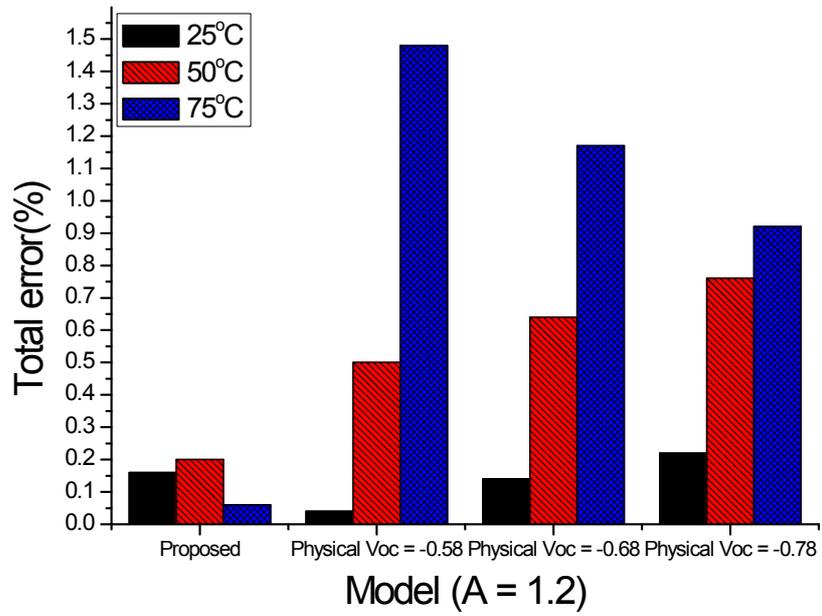
(a)



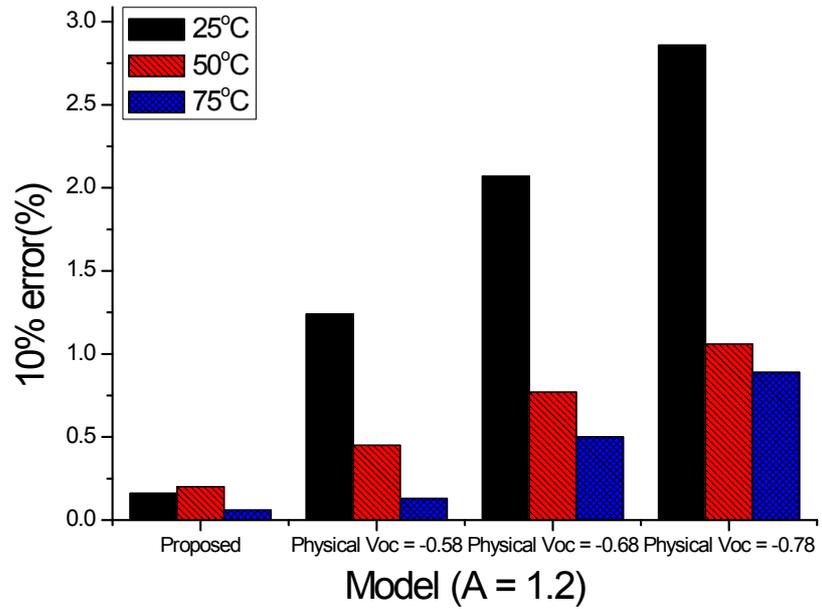
(b)

그림 4.13 제안모델과 PSIM 모델의 온도영향 비교

Fig. 4.13 Temperature dependency of the conventional and the proposed model



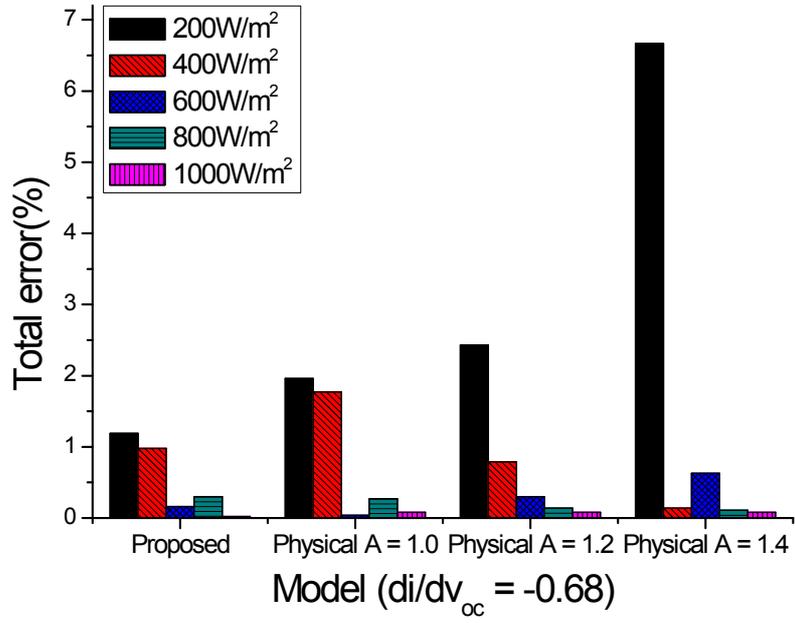
(c)



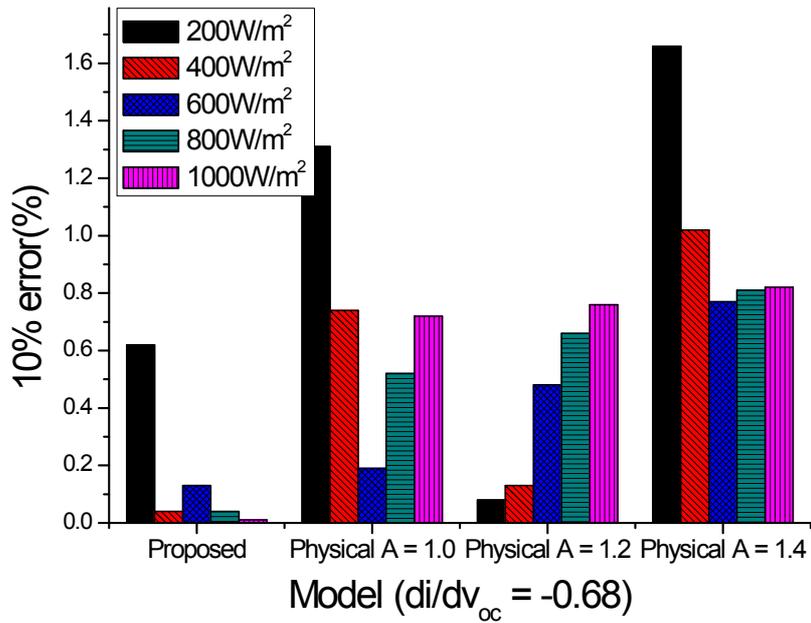
(d)

그림 4.13 제안모델과 PSIM 모델의 온도영향 비교 (계속)

Fig. 4.13 Temperature dependency of the conventional and the proposed model (continued)



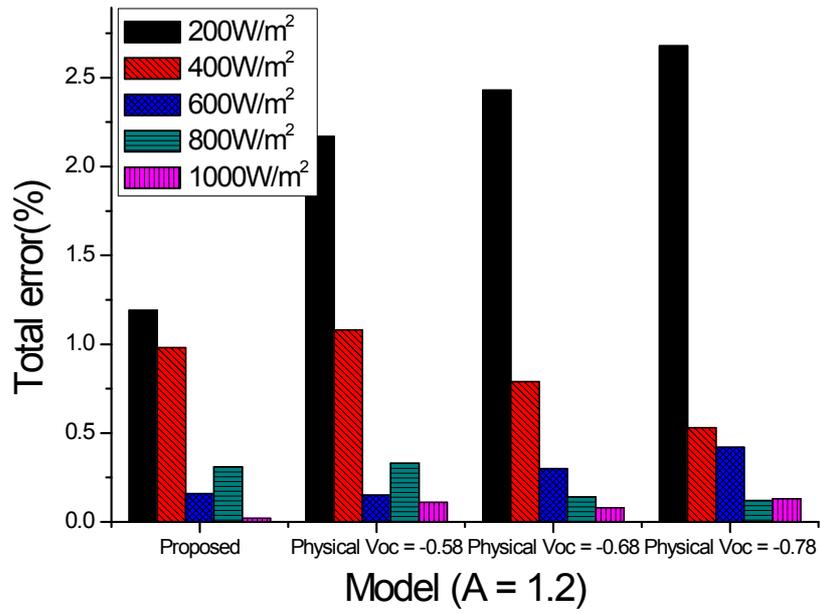
(a)



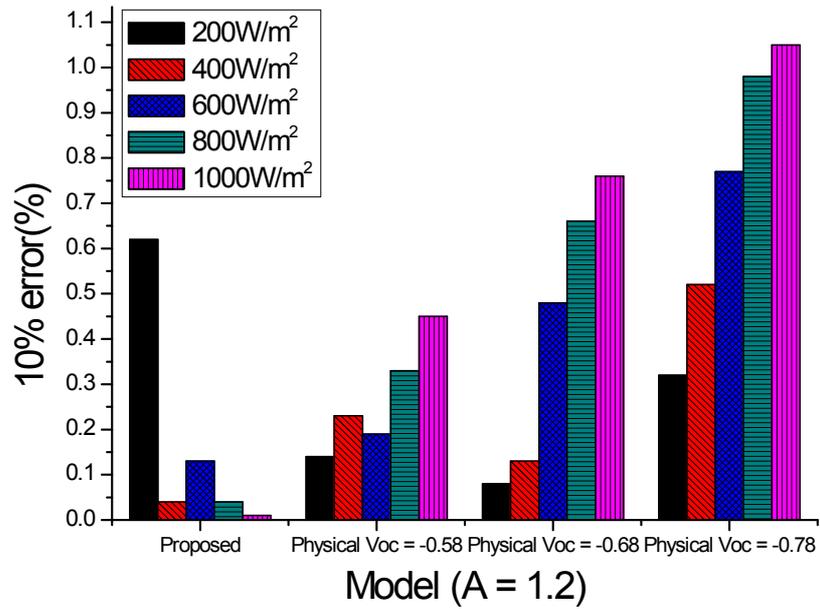
(b)

그림 4.14 제안모델과 PSIM 모델의 일사량영향 비교

Fig. 4.14 Irradiation dependency of the conventional and the proposed model



(c)



(d)

그림 4.14 제안모델과 PSIM 모델의 일사량영향 비교 (계속)

Fig. 4.14 Irradiation dependency of the conventional and the proposed model (continued)

우수한 경우가 발생하지만, 그 변화폭이 크고 제안모델은 사용자의 개입이 없어서 항상 일정한 출력결과를 제공하므로 후자의 경우가 신뢰도 면에서 더 우수하다고 할 수 있다. 다음으로 그림 4.14의 일사량의 변화에 따른 결과를 살펴보면, 온도변화에서의 결과와 마찬가지로 제안모델의 결과가 더 부정확한 경우가 존재한다. 하지만 전체 일사량의 환경변화에 대응하는 정도는 제안모델의 결과가 우수함을 확인 할 수 있으며, 이때 온도의 경우와 마찬가지로 사용자의 외부개입은 존재하지 않는다.

결과적으로 제안모델은 PSIM 모델과는 다르게 온도나 일사량의 변화에 대해서 더 정확한 모델을 제공하고 모델링 과정에서 사용자의 개입을 최소화하여 사용자에게 의해서 결정되는 파라미터의 영향을 줄이는 것을 통해서 기존보다 더 정확한 태양광패널 모델을 제공하는 것을 볼 수 있다.

4.3 제안방법의 하드웨어 응용

제안모델은 소프트웨어 구현과 마찬가지로 실제 하드웨어실험에 적용하기 위한 태양광 시뮬레이터 회로도 구현이 될 수 있다. 이때, 태양광 패널 시뮬레이터의 하드웨어는 DC

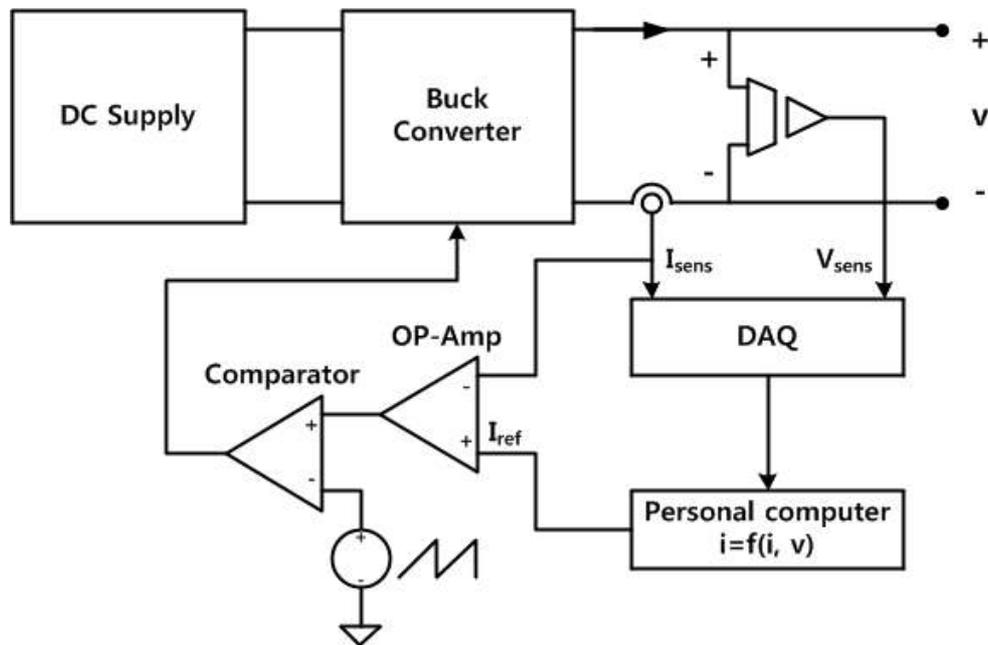


그림 4.15 하드웨어 회로의 개념도

Fig. 4.15 Schematic of hardware circuit

서플라이, 벡-컨버터, 전류제어 회로와 제안모델을 이용해서 구현이 가능하며, 하드웨어 회로의 개념도는 그림 4.15와 같이 나타낼 수 있다. 그림에서 기준전류, I_{ref} 를 생성하기 위해서 제안모델이 사용되는데 빠른계산을 위해서 PC를 사용하는 것이 적절하다고 판단된다. 이를 위해서 DAQ장비를 통해서 벡-컨버터의 출력전압 및 전류를 측정해서 통신으로 정보를 전달하며, 전달된 측정값을 통해서 제안모델이 I_{ref} 를 생성한다. 생성된 I_{ref} 는 아날로그 제어기를 통해서 벡-컨버터의 Duty를 조절해서 태양광 패널의 출력을 모사하게 된다.

그림 4.15를 기초로 본 절에서는 하드웨어응용을 위한 벡-컨버터의 설계 및 모델링을 하고 전류제어기를 설계하여 전체회로의 시뮬레이션을 통해서 검증한 뒤에 최종적으로 하드웨어로 구현하는 것을 목표로 한다.

4.3.1 벡-컨버터 전력단 설계

제안모델의 하드웨어구현을 위해서 본 논문에서는 전류로 제어되는 벡-컨버터를 이용하였다. 이때, 벡-컨버터의 설계사양은 표 4.5와 같이 정하였다. 벡-컨버터의 입력은 DC 서플라이로부터 50V로 고정된 전압값을 얻으며, 전압의 출력은 실제 태양광 패널인 KC125GT의 단락전압 0V로부터 개방전압인 21.7V까지 모사가 가능하며, 전류의 출력은 단락전류인 8.0A부터 개방전류인 0A까지 구현이 되도록 목표를 설정하였다. 이때, 하드웨어의 출력은 태양광 패널을 특성을 따라서 비선형적인 형태를 나타낼 것이다. 그리고 스위칭 주파수는 100kHz로 설정하였으며, 설정한 입력값과 출력전압 및 전류의 마진을 고려해서 인덕터와

표 4.5 벡-컨버터의 설계사양

Table 4.5 Specification of the buck converter

Specification	Value	Unit
Input voltage	50	V
Output voltage	0~21.7	V
Output current	0~8	A
Inductor	136	uH
Output capacitor	330	uF
Switching frequency	100	kHz

캐패시터는 각각 136 μ H와 330 μ F으로 설계하였다. 이때, 캐패시터의 값에 따라서 회로 전체의 반응이 결정되므로, 너무 크지도 않고 적절한 330 μ F을 선정하였다.

4.3.2 벽-컨버터 소신호 모델링

벽-컨버터의 제어기를 설계하기에 앞서 벽-컨버터의 소신호 모델링을 할 필요가 있다. 이는 소신호 모델을 통해서 얻은 벽-컨버터의 전달함수를 이용하면 적절한 제어기를 선택하고 설계할 수 있기 때문이다.[30] 먼저, 벽-컨버터가 사용되는 범위를 전압을 통해서 살펴보면, 최소 0V에서 최대 21.7V의 범위에서 제어가 필요함을 알 수 있다. 마찬가지로 전류의 경우 최소 0A에서 최대 8A의 범위에서 사용이 된다. 즉, 동작범위의 극점은 개방전압 V_{oc} 부근과 단락전류 I_{sc} 부근임을 알 수 있다. 양 끝점에서의 제어가 안정적이면 MPP에서의 제어도 안정적으로 될 것이다.

앞서 설정된 동작범위를 이용해서 벽-컨버터의 모델링을 진행할 때, 동작점은 가변부하에 의해서 결정된다. 즉, 부하에 따라서 출력의 전압 및 전류가 변하며, 모델링을 위해서는

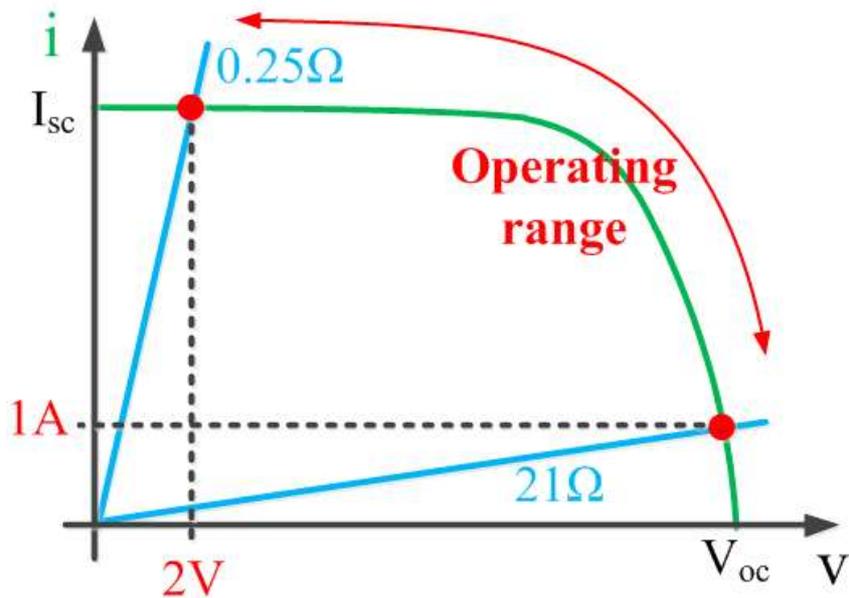


그림 4.16 태양광 시뮬레이터의 동작범위

Fig. 4.16 Operating range of PV simulator

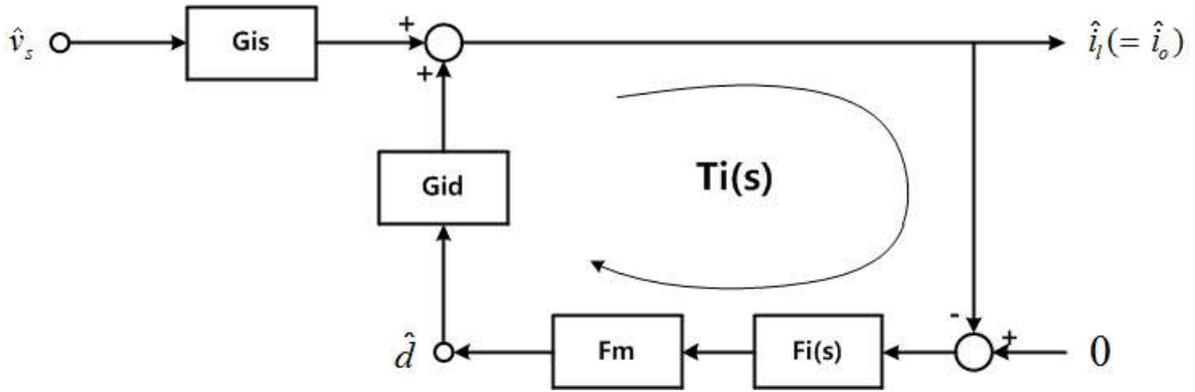


그림 4.17 전류제어의 제어흐름도

Fig. 4.17 Control loop of the system

V_{oc} 와 I_{sc} 에서의 부하를 결정할 필요가 있다.[31] 기본적으로 V_{oc} 에서는 부하저항이 매우 크며, I_{sc} 에서는 매우 작다고 할 수 있으나 설계를 위해서 그림 4.16과 같이 특정범위 안으로 동작범위를 설정하였다. 이때, V_{oc} 부근의 동작지점은 1A에 21V 지점이며, I_{sc} 의 부근은 2V에 8A지점으로 설정하였고 그때 부하는 그림과 같이 각각 21Ω과 0.25Ω이 된다. 이 두 동작점을 이용해서 벽-컨버터의 양끝 동작지점을 구할 수 있다.

양끝 동작지점을 이용해서 제어기의 설계를 진행하기 전에 벽-컨버터의 소신호 모델링을 하면, 아래의 전달함수를 얻을 수 있다.

$$G_{is} = \frac{\hat{i}_L(s)}{\hat{v}_s(s)} = \frac{D}{R} \frac{1+sCR}{1+s\frac{L}{R}+s^2LC} \quad (4.18)$$

$$G_{id} = \frac{\hat{i}_L(s)}{\hat{d}(s)} = \frac{V_s}{R} \frac{1+sCR}{1+s\frac{L}{R}+s^2LC} \quad (4.19)$$

이 식은 기본적인 벽-컨버터의 입력전압 대 인덕터전류 및 듀티 대 인덕터전류의 전달함수를 구한 것이다. 하지만 실제회로에서는 인덕터 및 캐패시터의 기생저항을 고려해야 하며, 이를 포함하는 최종식은 아래와 같이 구해진다.

$$G_{is} = \frac{\hat{i}_L(s)}{\hat{v}_s(s)} = \frac{D}{R+r_l} \frac{1+sC(R+r_l)}{1+s\left(\frac{L+C(R+r_c)r_l+r_cCR}{R+r_l}\right)+s^2\frac{LC(R+r_c)}{R+r_l}} \quad (4.20)$$

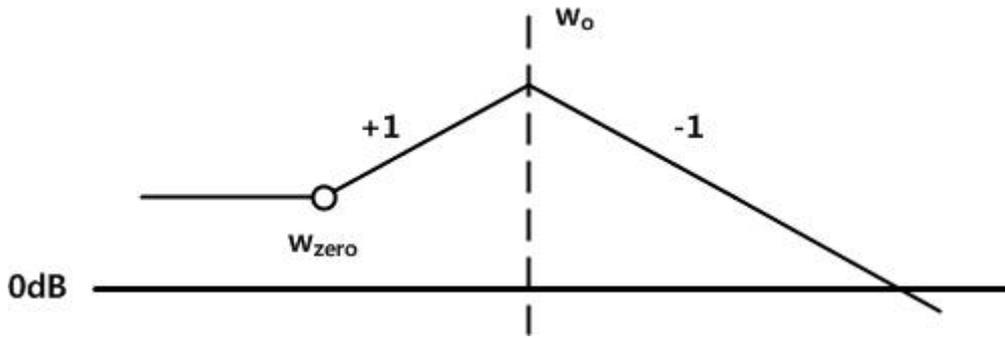
$$G_{id} = \frac{\hat{i}_L(s)}{\hat{d}(s)} = \frac{V_s}{R+r_l} \frac{1+sC(R+r_l)}{1+s\left(\frac{L+C(R+r_c)r_l+r_cCR}{R+r_l}\right)+s^2\frac{LC(R+r_c)}{R+r_l}} \quad (4.21)$$

식 (4.20)과 (4.21)에서 인덕터의 기생성분은 r_l 로, 캐패시터의 기생성분은 r_c 로 표현하였다. 그리고 벡-컨버터의 전류로부터 스위칭 주파수를 제어하기 위해서는 그림 4.17과 같은 제어회로를 구성해야 한다. 그림에서 제어기를 설계하기 위해서는 G_{id} 를 더 분석할 필요가 있으며, G_{id} 는 그림 4.18에서와 같이 점근적인 형태로서 표현이 가능하다. 점근선을 이용하면 시스템의 특성을 파악하는데 더 쉽기 때문에 본 논문에서는 점근적 분석을 이용해서 모델의 전달함수를 분석하고 제어기를 설계하였다.

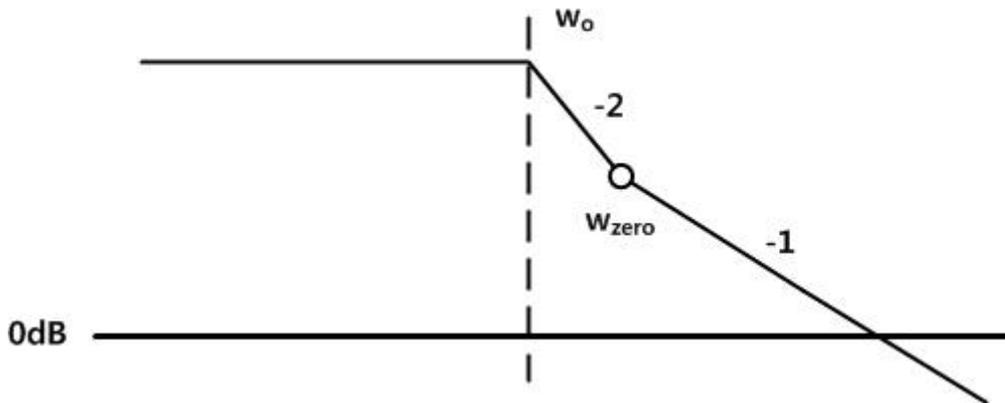
4.3.3 제어기 설계전략

그림 4.18에서 (a)는 V_{oc} 에서의 전달함수를 점근적으로 표현한 것이다. 여기서 중복된 pole에 의해서 w_o 지점에서 peaking이 발생하며, pole의 앞 쪽에 zero가 위치하는 것을 볼 수 있다. 그리고 (b)에서는 I_{sc} 부근에서의 전달함수를 확인 할 수 있는데, 여기서 더블 pole의 위치는 V_{oc} 의 경우와 같지만 zero의 위치가 pole의 바로 뒤로 이동한 것을 확인 할 수 있다. 결국, 그림과 같이 제안하는 벡-컨버터의 모델에서 zero의 위치가 이동하기 때문에 이를 고려한 제어기의 설계가 필요하다.

그림 4.18의 (b)에서 zero의 위치가 조금 더 높은 주파수로 이동한다면, -40dB/dec의 기울기 때문에 차단주파수 (cut-off frequency, f_c)를 설정하는 것이 매우 어렵다. 이 경우 그림 4.19와 같은 type3의 PID 제어기를 사용해서 두 개의 zero 중에서 하나는 -40dB/dec를 상쇄할 목적으로 사용할 수 있으나, type3를 사용할 경우 그림 4.19에서 보이듯이 V_{oc} 조건에서 차단주파수가 높은 주파수 대역으로 이동하기 때문에 이를 설정하는 것이 어려울 수 있다. 본 논문에서는 제어기의 zero 하나를 적절히 위치하면, 추가적인 zero를 사용하는 것이 필요가 없다고 판단하여 하나의 pole과 하나의 zero를 가지는 type2의 PI제어기를 사용하였다. 이때, PI제어기의 zero의 위치는 V_{oc} 에서의 벡-컨버터의 zero와 더블 pole의 사이에 위치



(a) V_{oc} condition

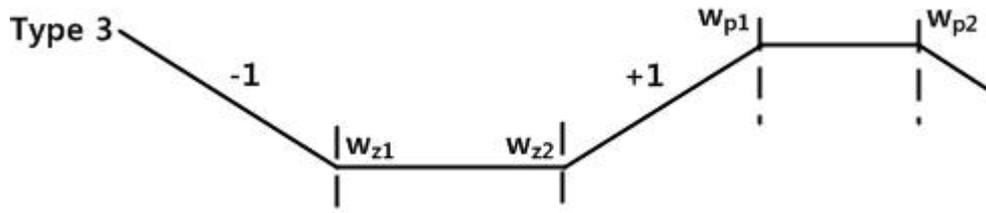


(b) I_{sc} condition

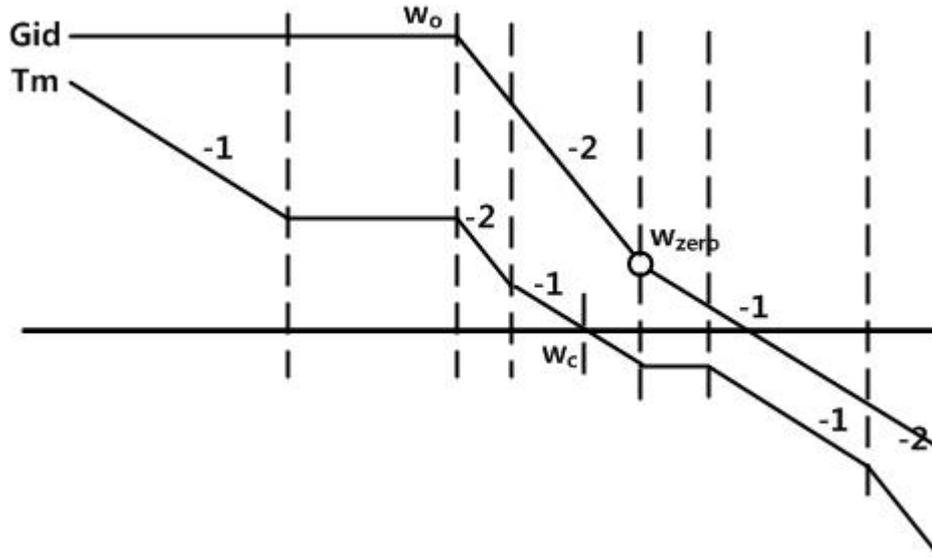
그림 4.18 점근적 전달함수

Fig. 4.18 asymptotic transfer function

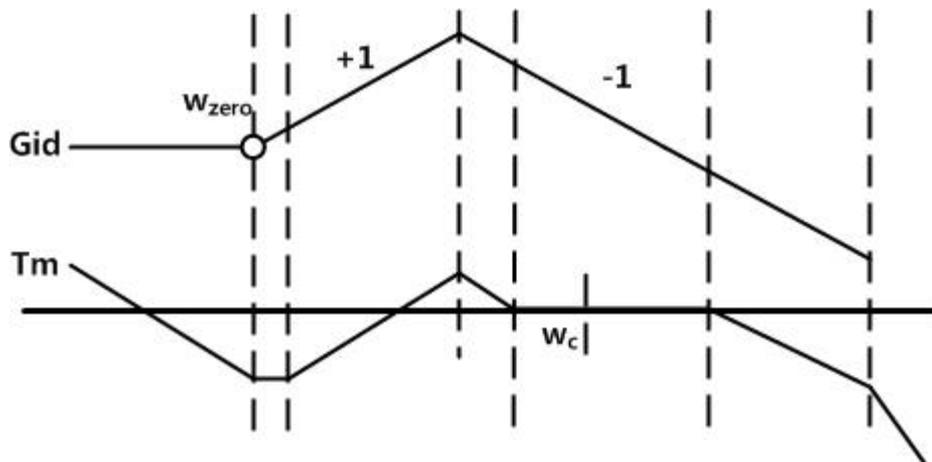
시켰고 type2 PI 제어기를 이용한 벡-컨버터의 점근적인 Bode plot은 그림 4.20와 같이 나타낼 수 있다. 최종적인 목표는 차단주파수를 20kHz 부근에 두는 것이며, 그때 충분한 위상여유를 갖도록 설계하는 것이다. 이를 MATLAB을 이용해서 확인해보면, 그림 4.21과 같이 나타난다. 그림에서 확인할 수 있듯이 제어기를 통해서 충분한 위상여유를 가지는 것을 확인할 수 있고 차단주파수 또한 목표대로 20kHz의 부근에 위치하는 것을 확인할 수 있다.



(a) Type3 제어기



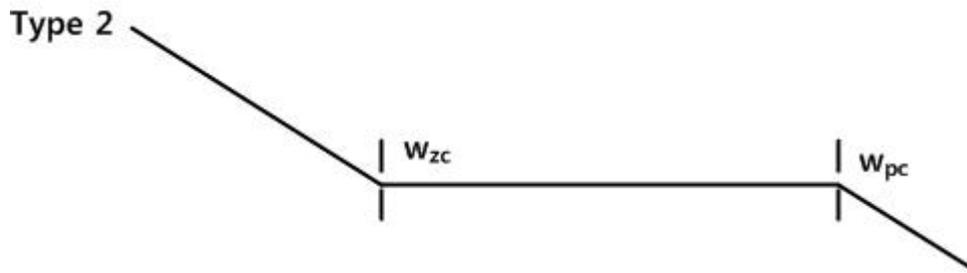
(b) transfer function at V_{oc} condition



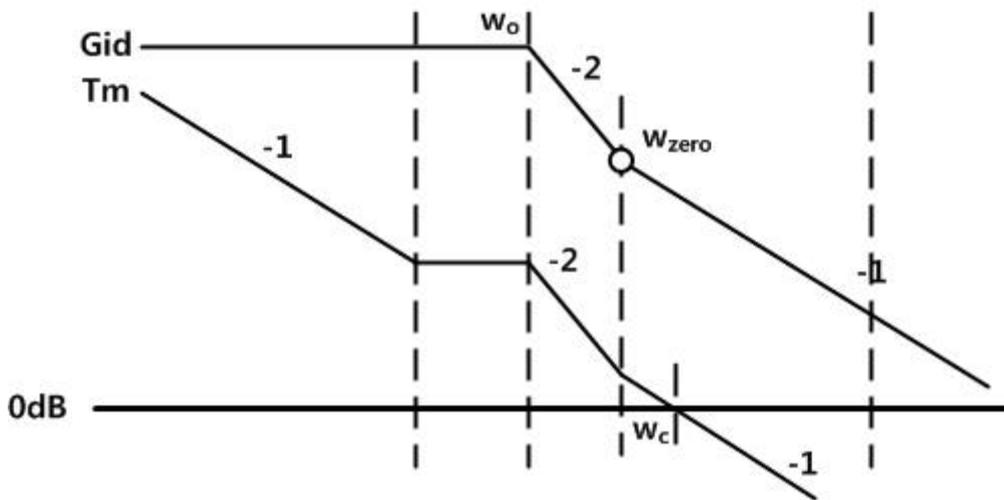
(c) transfer function at I_{sc} condition

그림 4.19 Type3 제어기

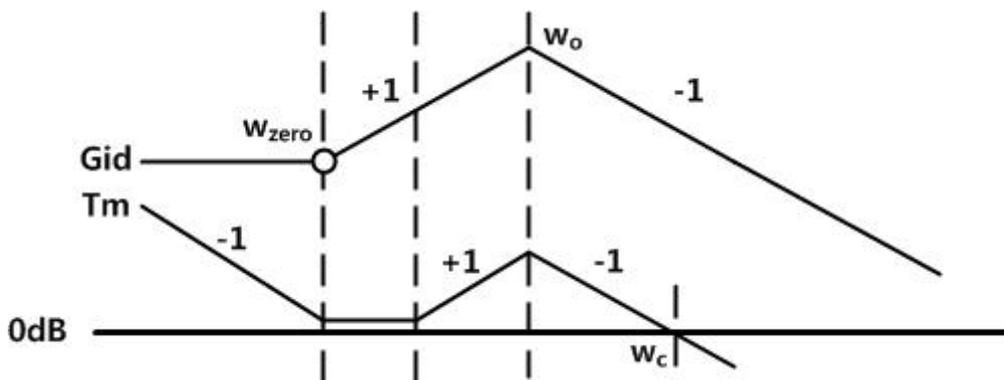
Fig. 4.19 Type3 Compensator



(a) Type2 제어기



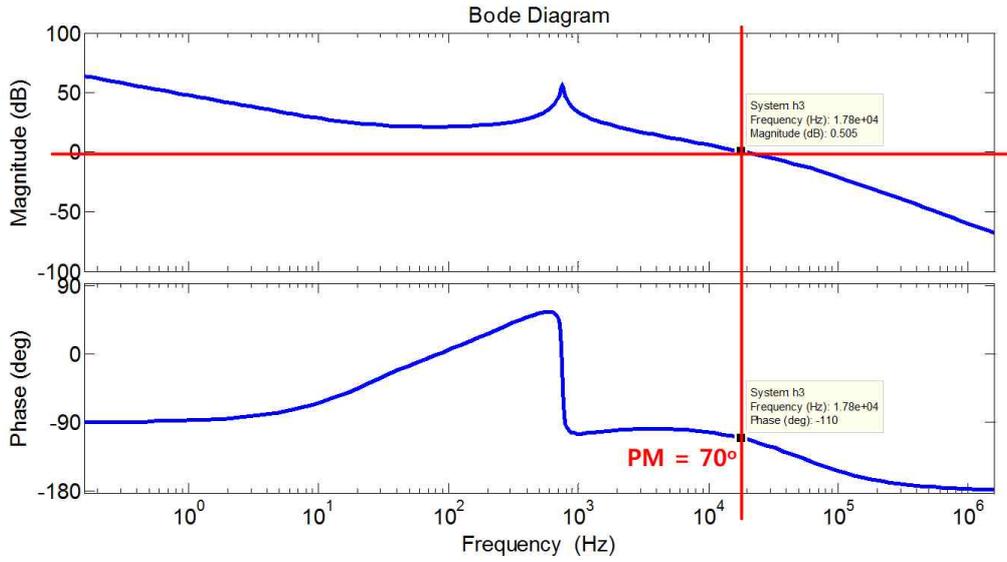
(b) transfer function at V_{oc} condition



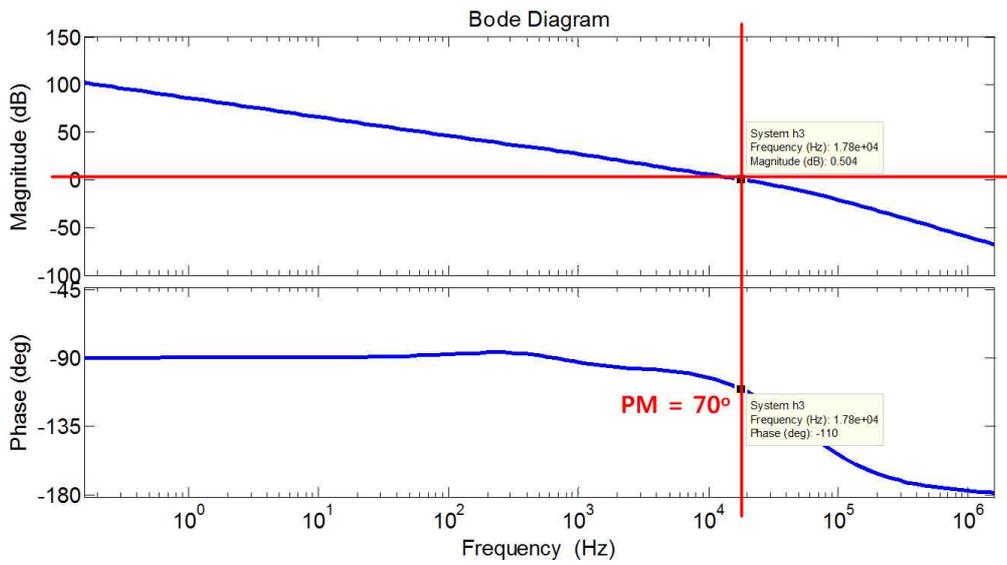
(c) transfer function at I_{sc} condition

그림 4.20 Type2 제어기

Fig. 4.20 Type2 Compensator



(a) V_{oc} condition



(b) I_{sc} condition

그림 4.21 MATLAB 검증

Fig. 4.21 Verification using MATLAB

4.3.4 전류 제어기의 설계

앞선 절에서 구한 type2 PI제어기는 그림 4.22과 같이 구성할 수 있으며, 제어기를 구성하기 위한 각각의 저항 및 캐패시터의 값은 아래의 식으로부터 계산할 수 있다.

$$F_v = \frac{Z_2(s)}{Z_1(s)} = \frac{K_v \left(1 + \frac{s}{w_{zc}}\right)}{s \left(1 + \frac{s}{w_{pc}}\right)} \quad (4.22)$$

$$K_v = \frac{1}{R_1(C_2 + C_3)} \quad (4.23)$$

$$w_{zc} = \frac{1}{R_2 C_2} \quad (4.24)$$

$$w_{pc} = \frac{1}{R_2 \left(\frac{C_2 C_3}{C_2 + C_3}\right)} \quad (4.25)$$

계산된 저항 및 캐패시터의 값은 표 4.6에 정리하였다. 이때, 설계값은 $C_2 \gg C_3$ 의 조건을 만족해야 하며, 표를 통해서 조건을 만족함을 확인할 수 있다. 표의 값을 이용해서 type2의 PI제어기를 구성할 수 있으며, 그림 4.22의 I_{ref} 는 제안모델을 통해서 공급해 줄 수 있다. 최종적인 구현 및 회로는 다음의 절에서 소개한다.

표 4.6 제어기의 설계사양

Table 4.6 Specification of the current controller

Parameter	Value	Unit
R_1	2	k Ω
R_2	2.7	k Ω
C_2	200	nF
C_3	1	nF

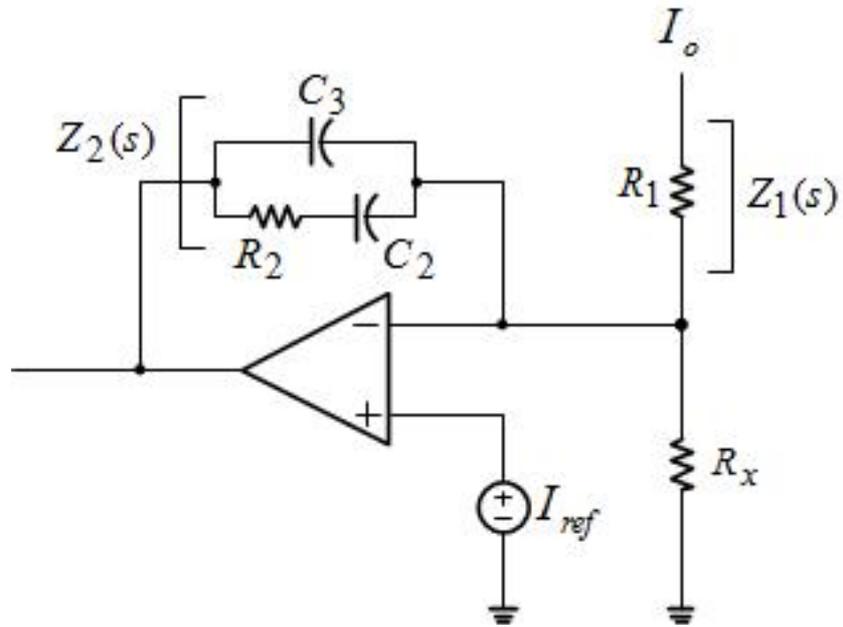


그림 4.22 Type2 제어기의 구성

Fig. 4.22 Structure of type2 compensator

4.3.5 제어기 시뮬레이션 검증

PSIM 시뮬레이션으로 결과를 확인하기 위해서 설계한 벽-컨버터, type2 PI제어기와 제안 모델을 이용해서 그림 4.23의 PSIM회로를 구성할 수 있다. 그림에서 DC 서플라이는 전압원으로 구현하였으며, 태양광 패널 시뮬레이터를 위한 벽-컨버터가 PSIM으로 구현되었다. 그리고 C block으로 구현한 제안모델은 벽-컨버터의 출력전압과 전류를 측정한 뒤에 기준 전류, I_{ref} 를 생성해서 OP-Amp에 공급해준다.

제안하는 하드웨어회로가 제어가 잘 되는지 확인하기 먼저 각각의 양끝의 동작지점에서 I_{ref} 를 조금씩 변화시켰고 그때의 결과는 그림 4.24와 같이 나타난다. 그림에서 확인할 수 있는 것은 하드웨어로 구현된 제안모델의 I_{ref} 가 변할 때, 전류가 잘 제어되는 것을 확인할 수 있다. 다음으로 기준전류가 일정할 때, 부하저항이 변할 경우에 제어가 제대로 되는지 확인하였고 그 결과는 그림 4.25와 같다. 그림에서 기준전류는 각각 V_{oc} 에서 1A, I_{sc} 에서 8A로 설정이 되었다. 그리고 저항은 V_{oc} 에서 21Ω에서 23Ω으로 변화였고 I_{sc} 에서 0.25Ω에서 0.27Ω으로 변화도록 설정하였다. 그 결과 부하가 변하는 순간 전류가 바뀌지만 이내 기준 전류인 1A 및 8A의 값으로 제어가 되는 것을 확인할 수 있다.

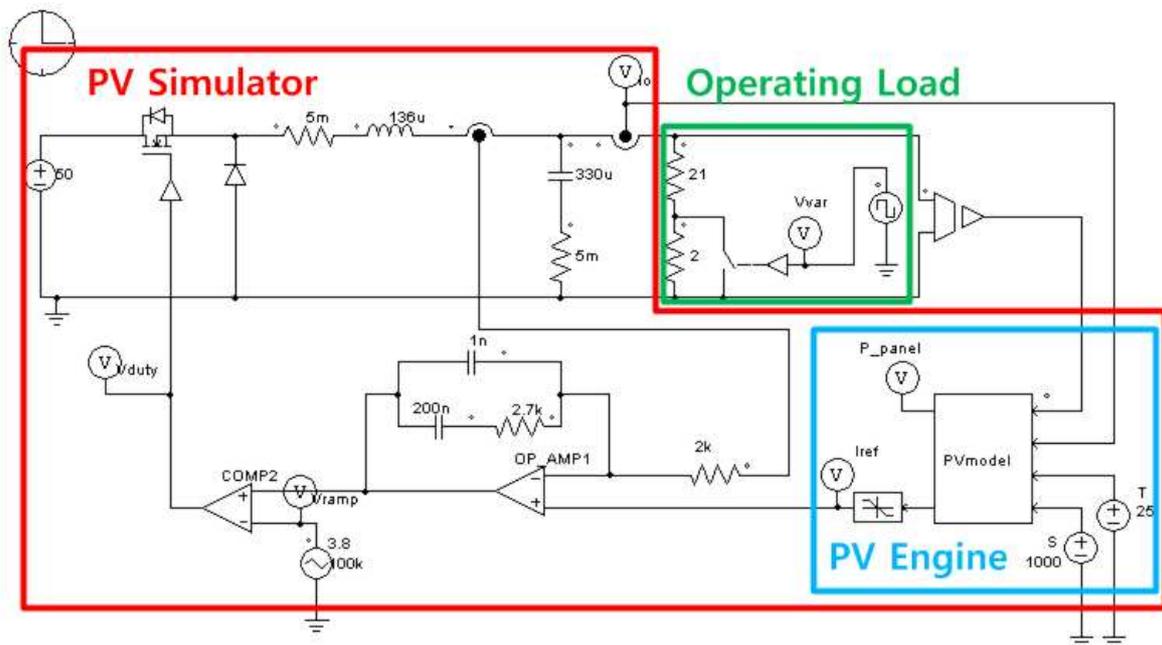
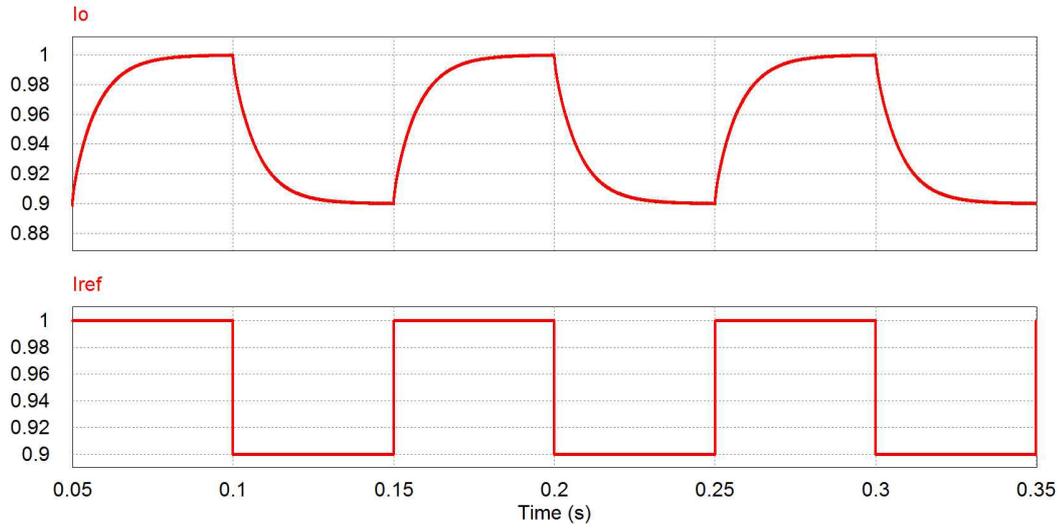
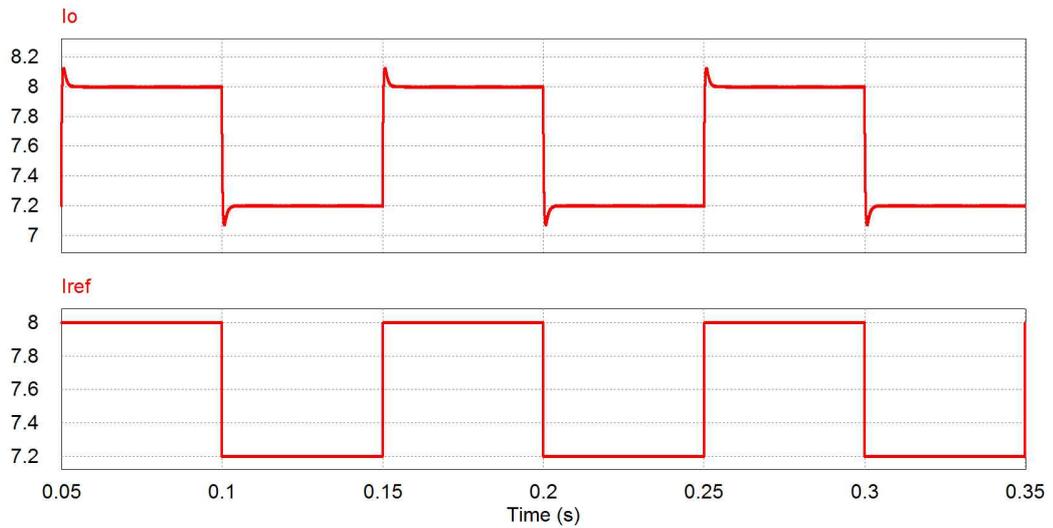


그림 4.23 제안모델의 하드웨어 시뮬레이션 회로

Fig. 4.23 Simulation circuit of the proposed model



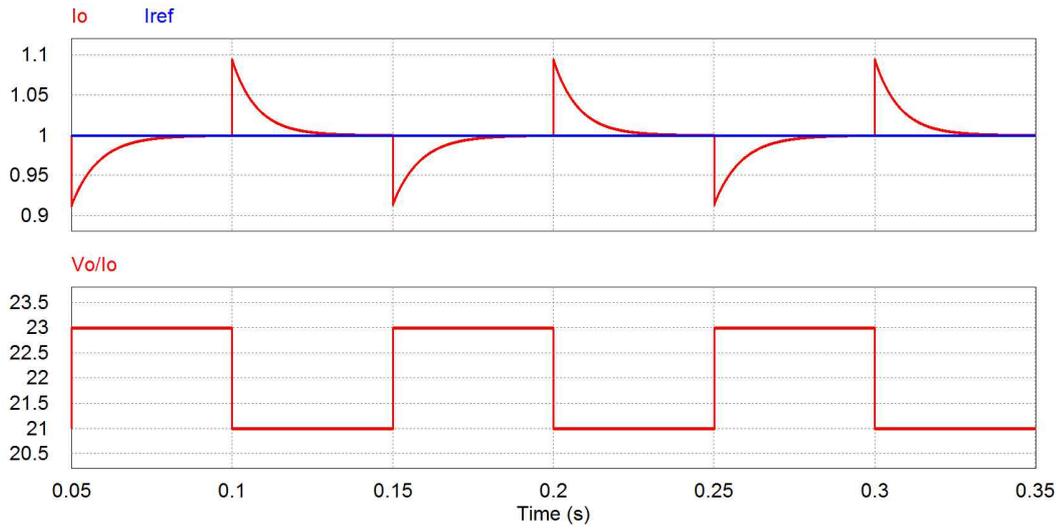
(a) V_{oc} condition



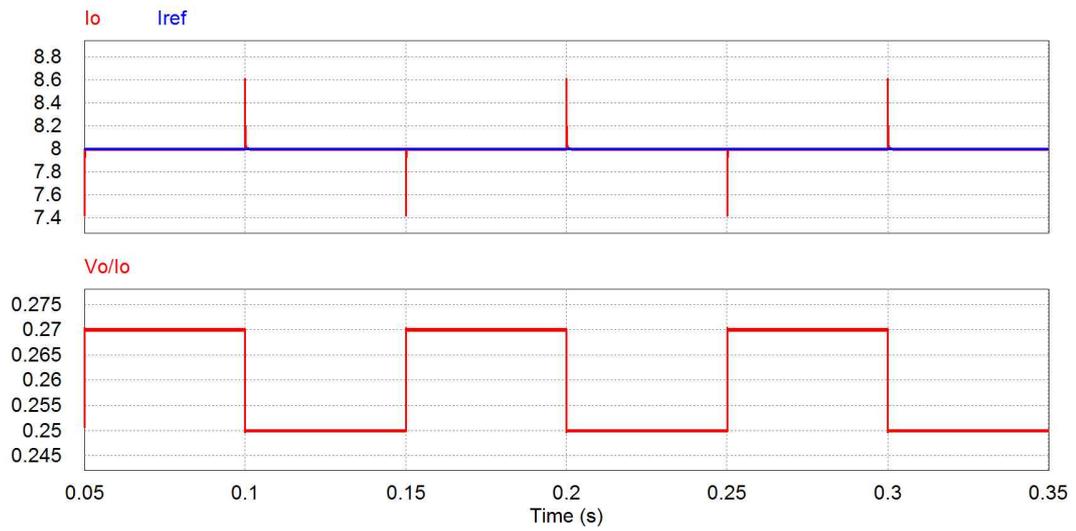
(b) I_{sc} condition

그림 4.24 기준지령에 대한 동적응답

Fig. 4.24 Dynamic response of step reference current



(a) V_{oc} condition



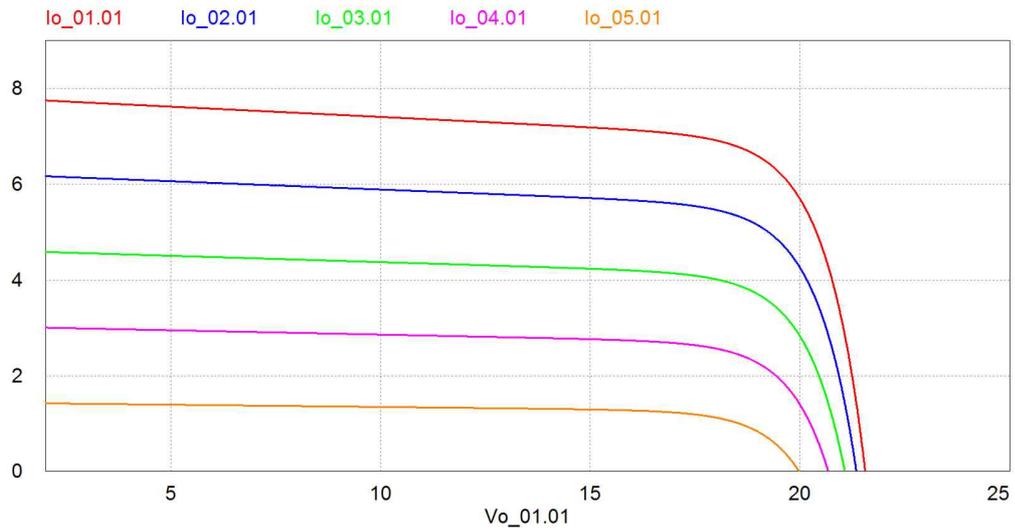
(b) I_{sc} condition

그림 4.25 부하외란에 대한 동적응답

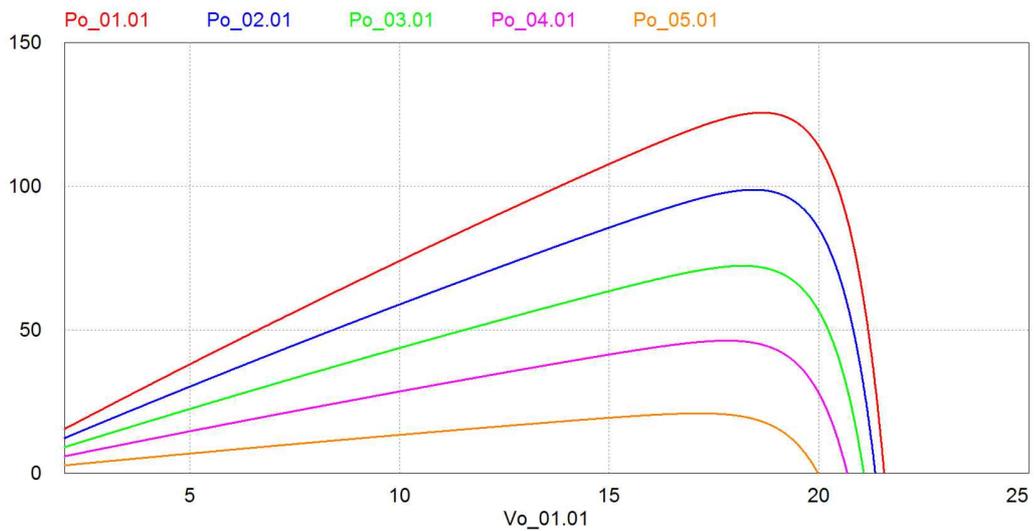
Fig. 4.25 Dynamic response of step load disturbance

4.3.6 PV 특성모사 성능평가

이 결과를 바탕으로 실제 태양광 패널인 KC125G에 대해서 시뮬레이션을 진행하였다. 우선, 그림 4.26은 KC125GT의 일사량에 대한 특성 그래프로서 일사량을 $200\text{W}/\text{m}^2$ 에서



(a) Current

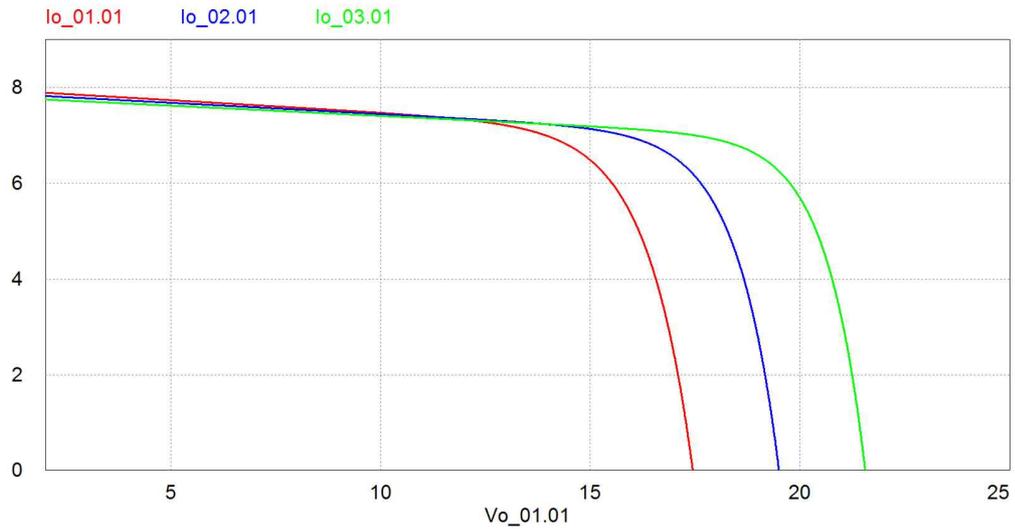


(b) Power

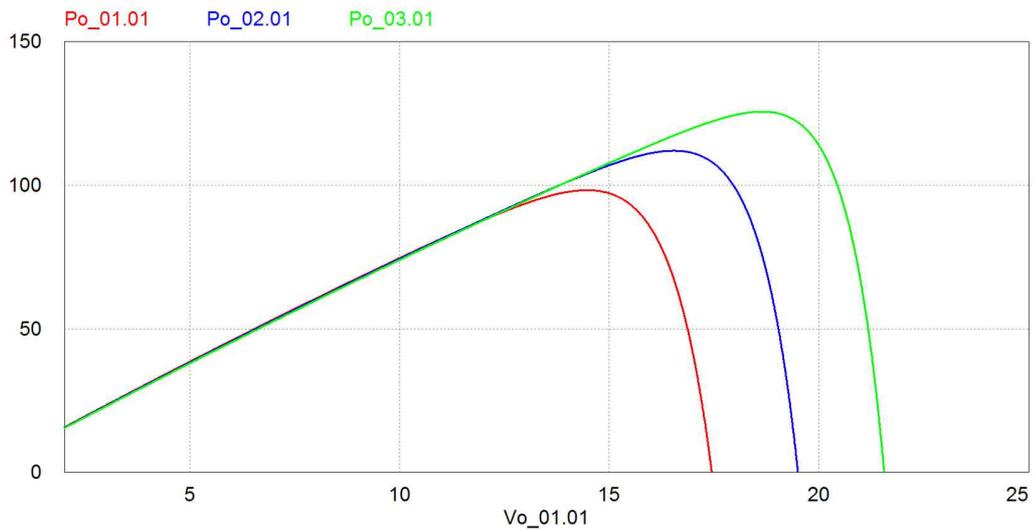
그림 4.26 KC125GT의 일사량에 대한 I-V 및 P-V 특성곡선

Fig. 4.26 I-V and P-V Characteristic of KC125GT under irradiation

1000W/m²으로 200W/m²씩 증가시키면서 I-V 및 P-V커브를 추출한 것이다. 마찬가지로 온도변화에 대한 제안모델의 시뮬레이션 결과를 그림 4.27에 나타냈다. 그림 4.26과 4.27의 결과는 제안모델이 KC125GT 패널의 일사량과 온도에 대한 특성을 잘 모사할 수 있음을 보여준다. 이를 바탕으로 변하는 일사량 및 온도에 대해서 제안모델의 출력변화를 그림 4.28과 4.29로 나타낼 수 있으며, 이때 그림 4.28의 경우 일사량의 변화에 대해서 보여주



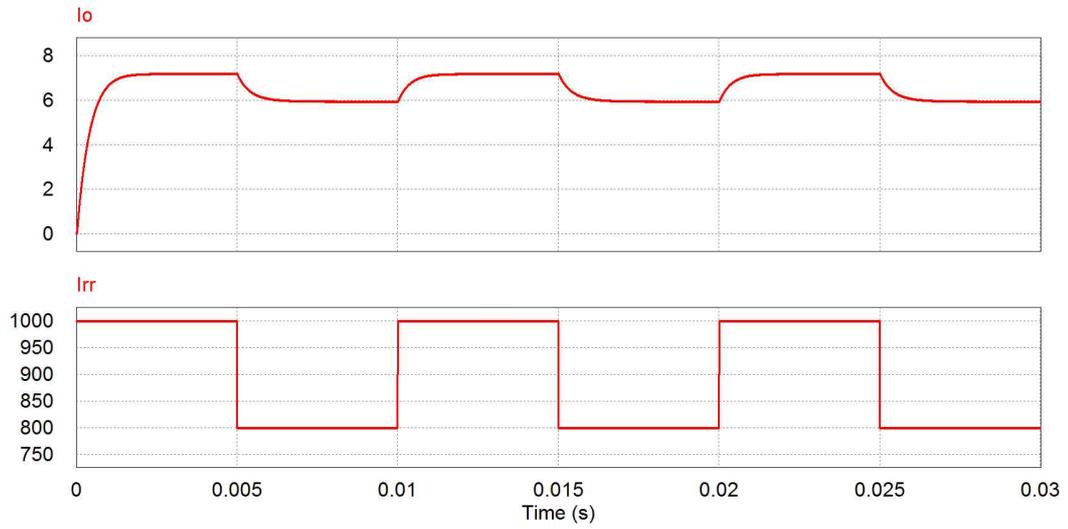
(a) Current



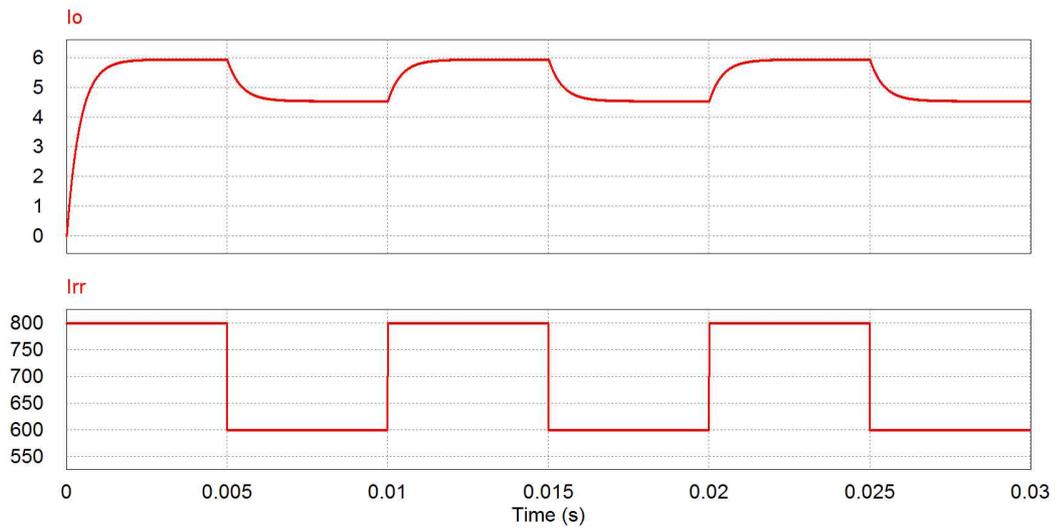
(b) Power

그림 4.27 KC125GT의 온도에 대한 I-V 및 P-V 특성곡선

Fig. 4.27 I-V and P-V Characteristic of KC125GT under temperature



(a) 800 to 1000 W/m²



(b) 600 to 800 W/m²

그림 4.28 제안모델의 일사량에 대한 응답특성

Fig. 4.28 Response of the proposed model under irradiation

며, 그림 4.29는 온도변화에 대한 제안모델의 반응을 보여준다. 결과를 통해서 제안모델은 태양광 PCS의 성능평가 또는 MPPT 알고리즘의 개발과 같은 분야에도 응용이 가능한 것을 확인할 수 있다.

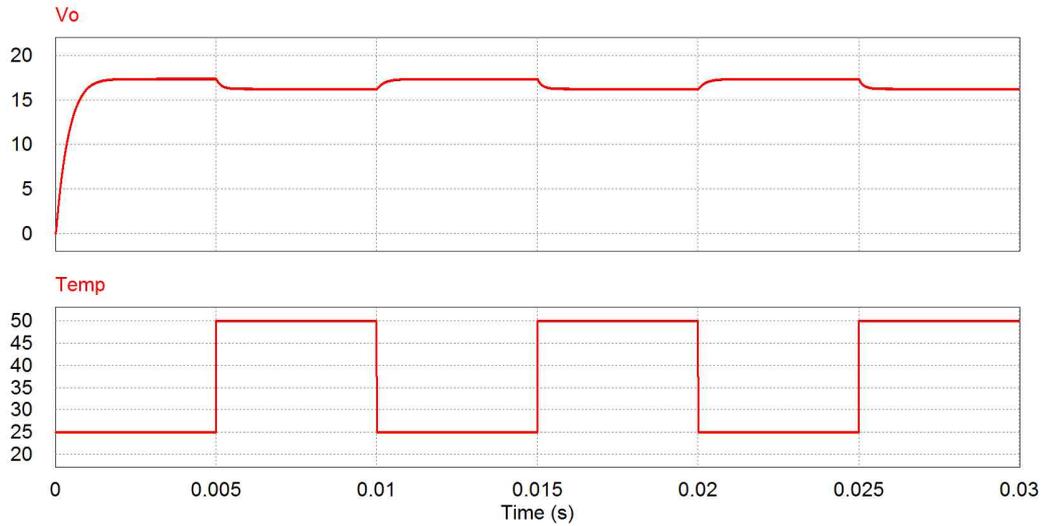


그림 4.29 제안모델의 온도에 대한 응답특성 (25 to 50 °C)

Fig. 4.29 Response of the proposed model under temperature (25 to 50 °C)

5. 결 론

본 논문에서는 태양광 시스템의 개발과정에서 많이 쓰이고 있는 태양광 패널의 등가회로 모델에 대해서 새로운 모델을 제안하고 이 모델의 소프트웨어 및 하드웨어 구현에 대해서 연구하였다. 본문에서 기존의 태양광 패널의 모델링 방법에 대해서 분석하고 장점 및 단점을 정리하였고 기존의 단점을 극복하기 위해서 제안방법은 최적화 방법을 이용하였다. 그 결과 제안방법은 기존보다 더 정확한 태양광 패널 모델을 제공함을 확인할 수 있었고 제안모델은 실제 패널의 종류에 관계없이 신뢰도 또한 보장하는 모델이다. 비교 및 검증의 타당한 근거를 확보하기 위해서 본 논문에서는 EN50530의 유럽규격을 사용하였다.

제안모델의 사용성 및 실제 응용을 위해서 PSIM을 통한 소프트웨어 모델을 구현하였으며, 그 결과 PSIM의 C block만을 이용해서 C언어로 제안모델을 구현 할 수 있었다. 이때, 제안모델의 타당성은 기존의 PSIM에서 제공하는 물리적모델과의 비교를 통해서 진행하였고 EN50530의 규격이 다시 사용되었다. 기존의 PSIM의 모델은 외부에서 사용자가 파라미터를 직접 결정해야 했기에 사용하기에 불편한 점이 있었고 사용자의 선택에 따라서 모델의 오차가 발생하여 신뢰도를 보장하기도 어려운 모델이었다. 하지만 제안모델은 데이터시트의 값만을 이용해서 태양광 패널의 모델링을 진행할 수 있었고 사용자의 개입을 최소화하여 모델의 정확도 및 신뢰도를 확보할 수 있었다.

마지막으로 제안모델은 실제 하드웨어의 PV 시뮬레이터의 엔진으로도 사용될 수 있으며, 이때 제안모델은 하드웨어 회로의 기준전압 또는 전류를 결정하는 역할을 한다. 이는 PSIM을 통한 시뮬레이션을 통해서 검증하였으며, 향후 하드웨어 회로로도 구성할 수 있다. 시뮬레이션의 결과는 전류제어 회로를 이용한 하드웨어가 제안모델이 발생시키는 기준전류의 값을 잘 추종할 수 있으며, 일정한 기준전류 PV 특성을 잘 모사함을 확인하였다.

결과적으로 본 논문에서 제안된 새로운 태양광 패널 모델은 좋은 정확도와 기존보다 향상된 신뢰도를 제공할 수 있으며, 소프트웨어에서의 PV 모델블럭뿐 아니라 하드웨어의 실험단계에서도 PV 시뮬레이터로서 유용하게 쓰일 수 있음을 증명하였다. 이를 통해서 제안모델의 사용폭이 매우 넓음을 확인할 수 있다. 향후에는 실제 PV 시뮬레이터를 구현하여 시뮬레이션 결과를 보완할 필요가 있을 것이다.

참 고 문 헌

- [1] J. Cubas, S. Pindado, and M. Victoria, "On the analytical approach for modeling photovoltaic systems behavior," *Journal of Power Source*, Vol. 247, pp. 467-474, Feb., 2014.
- [2] Z. Ouennoughi and M. Chegaar, "A Simpler Method for Extracting Solar cell Parameters using the Conductance Method," *Solid-State Electronics*, Vol. 43, Issue 11, pp.1985-1988, 1999.
- [3] M. Chegaar, Z. Ouennoughi, and A. Hoffmann, "A new method for evaluating illuminated solar cel parameters," *Solid-State Electronics*, Vol. 45, Issue 2, Feb., pp.293-296, 2001.
- [4] A. Wagner, "Photovoltaik Engineering: Die Methode der Effektiven Solarzellen-Kennlinie," Springer-Verlag, Berlin, Heidelberg, New York, 1999.
- [5] R. Kezzar, M. Zereg, and A. Khezzar, "Modeling improvement of the four parameter model for photovoltaic modules," *Solar Energy*, Vol. 110, pp. 452-462, Dec., 2014.
- [6] W. Xiao, W. G. Dunford, and A. Capel, "A Novel Modeling Method for Photovoltaic cells," *IEEE 35th Annual Power Electronics Specialists Conference (PESC)*, Vol. 3, pp. 1950 - 1956, Jun., 2004.
- [7] J. Crispim, M. Carreira, and R. Castro, "Validation of Photovoltaic Electrical Models against Manufacturer's Data and Experimental Results," *International Conference on Power Engineering, Energy and Electrical Drives*, pp. 556-561, Apr. 12-14, 2007.

- [8] M. G. Villalva, J. R. Gazoli, and E. R. Filho, "Comprehensive Approach to Modeling and Simulation of Photovoltaic Arrays," *IEEE Transactions on Power Electronics*, Vol. 24, Issue 5, pp. 1198-1208, May, 2009.
- [9] H. A. B. Siddique, P. Xu, and R. W. De Doncker, "Parameter Extraction Algorithm for One-Diode Model of PV Panels based on Datasheet Values," *International Conference on Clean Electrical Power (ICCEP)*, pp. 7-13, Jun. 11-13, 2013.
- [10] D. Sera, R. Teodorescu, and P. Rodriguez, "PV Panel Model Based on Datasheet Values," *IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 2392-2396, Jun., 2007.
- [11] D. Chan and J. Phang, "Analytical Methods for the Extraction of Solar cell single- and double-diode Model Parameters," *IEEE Transactions on Electron devices*, Vol. 34, Issue 2, pp. 286-293, Feb., 1987.
- [12] H. A. Park and H. S. Kim, "Mathematical Consideration on PV Cell Modeling," *The Transactions of Korean Institute of Power Electronics*, Vol. 19, No. 1, Feb., 2014.
- [13] L. Peng, Y. Sun, and Z. Meng, "An improved model and parameters extraction for photovoltaic cells using only three state points at standard test condition," *Journal of Power Source*, Vol. 248, pp. 621-631, Feb., 2014.
- [14] M. S. Ismail, M. Moghavvemi, and T. M. I. Mahlia, "Characterization of PV panel and global optimization of its model parameters using genetic algorithm," *Energy Conversion and Management*, Vol. 73, pp. 10-25, Sept., 2013.

- [15] K. Ishaque and Z. Salam, "An improved modeling method to determine the model parameters of photovoltaic (PV) modules using differential evolution (DE)," *Solar Energy*, Vol. 85, Issue 9, pp. 2349-2359, Sept., 2011.
- [16] X. Yuan, Y. Xiang, and Y. He, "Parameter extraction of solar cell models using mutative-scale parallel chaos optimization algorithm," *Solar Energy*, Vol. 108, pp. 238-251, Oct., 2014.
- [17] P. Venkataraman, *Applied Optimization with MATLAB Programming*, 2nd ed., WILEY, 2009.
- [18] M. G. Villalva, J. R. Gazoli, and E. R. Filho, "Modeling and Circuit-based Simulation of Photovoltaic Arrays," IEEE Brazilian Power Electronics Conference, pp. 1244-1254, Sept., 2009.
- [19] *PSIM User's Guide*, Ver. 9, May, 2010, POWERSIM.
- [20] IEC EN50530, Standard for Overall Efficiency of Photovoltaic Inverters, CENELEC, Stassart 35, B-1050 Brussels.
- [21] J.J. Soon and K. S. Low, "Photovoltaic Model Identification Using Particle Swarm Optimization With Inverse Barrier Constraint," *IEEE Transactions on Power Electronics*, Vol 27, Issue 9, pp. 3975-3983, Sept., 2012.
- [22] D. L. King, J. A. Kratochvil, and W. E. Boyson, "Temperature Coefficients for PV Modules and Arrays: Measurement Method, Difficulties, and Results," *IEEE 26th Photovoltaic Specialists Conference*, pp. 1183-1186, Sept., 1997.

- [23] W. De Soto S. A. Klein, and W. A. Beckman, "Improved and Validation of a Model for Photovoltaic Array Performance," *Solar Energy*, Vol. 80, Issue 1, pp. 78-88, Jan., 2007.
- [24] A. N. Celik and N. Acikgoz, "Modeling and Experimental Verification of the Operating current of mono-crystalline Photovoltaic Modules using four- and five-parameter Models," *Applied Energy*, Vol. 84, Issue 1, pp. 1-15, Jan. 2007.
- [25] C. Gonzalez-Moran, P. Arboleya, D. Reigosa, G. Diaz, and J. Gomez-Alexandre, "Improved Model of Photovoltaic Sources considering ambient Temperature and Solar Irradiation," *IEEE PES/IAS Conference on Sustainable Alternative Energy (SAE)*, pp. 1-6, Sept., 2009.
- [26] S. S. Raghuvanshi, A. Singh, and Y. Mokhariwale, "A Comparison & Performance of Simulation Tools MATLAB/SIMULINK, PSIM & PSPICE for Power Electronics Circuit," *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, Vol. 2, Issue 3, Mar., 2012.
- [27] PSIM Tutorial, *How to Use Solar Physical Model*, POWERSIM.
- [28] A. K. Singhal and R. Narvey, "PSIM and MATLAB based Simulation of PV Array for Enhance the Performance by using MPPT Algorithm," *International Journal of Electrical Engineering*, Vol. 5, Issue 5, pp. 511-520, 2011.
- [29] J. A. Gow and C. D. Manning, "Development of a photovoltaic array model for use in power-electronics simulation studies," *IEE Proceeding - Electric Power Applications*, Vol. 146, Issue 2, pp. 193-200, Mar., 1999.

[30] J. Zhao and J. W. Kimball, "A Digitally Implemented Photovoltaic Simulator with a Double Current Mode Controller," *IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 53-58, Feb., 2012.

[31] Y. Li, T. Lee, F. Z. Peng, and D. Liu, "A Hybrid Control Strategy for Photovoltaic Simulator," *IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 899-903, Feb., 2009.

부 록

MATLAB code

```

%% management functions
clear % clear all variable/information in the workspace - use
CAUTION
clc % position the cursor at the top of the screen
close % closes the figure window
format compact % avoid skipping a line when writing to the command
window
warning off % don't report any warnings like divide by zero etc.

lowbound = 0; intvl = 1; ntrials =20; tol = 0.001;
text1 = ['The function for which the minimum is sought must be' ...
'\n a MATLAB function M - File. Given a vector dependent ' ...
'\nvariable it must return a scalar value. This is the' ...
'\nfunction to be MINIMIZED. Please select function name' ...
'\nin the dialog box and hit return : \n '];

fprintf(text1) % displaying prompt
[file, path] = uigetfile('.\*.m','File of type MATLAB Script');

if isstr(file)
    functname = strrep(file, '.m', ''); % strip .m from file name
else

    fprintf('\n\n');
    text2 = ['You have chosen CANCEL or the file was not acceptable '
...
'\nThe program needs a File to Continue' ...
'\nPlease call Pattern Search again and choose a file OR '
...
'\npress the up-arrow button to scroll through previous
commands \n' ...
'\nBYE'];
    fprintf(text2);

```

```

end

clear text1 text2;
fprintf('\n');
string1 = strcat('The function you have chosen is :: ',funcname);
fprintf(string1);
clear string1;

Nmax = 1000;  epsfdif = 1.0e-8; epsxdif = 1.0e-8;

ok = 1;
while(ok)
    string1 = ['Input the starting design vector. This is mandatory'
    ...
    '\nas there is no default vector setup. The length of your'
    ...
    '\nvector indicates the number of unknowns. ' ...
    '\nPlease enter it now and hit return\n'] ;

    xdes = input(string1);
    fprintf('\n')
    fprintf(' The initial design vector [ %6.2f %6.2f ]',xdes);
    fprintf('\n')

    if (isempty(xdes) == 0)
        break;
    end
end

n = length(xdes);                                % length of design vector

% initialize the search vector to zero
for i = 1:n;
    searchi(i) = 0.0;
end

%      start      the      search      for      Nmax      iterations
-----
% In this method for each iteration there will be n + 1 iterations

```

```

% the first n iterations correspond to the unit vector search
directions
% the n+1 iteration correspond to the summ of the previous
displacements
%-----
-----
iders = 1; % iteration counter
idersall = 0; % iteration counter

% store values
xstore(iders,:) = xdes;
fstore(iders) = feval(funcname,xdes);
astore(iders) = lowbound;

countloop = 0;
for outloop = 1:Nmax; % loop on cycles
    xbegin = xdes;
    fbegin = feval(funcname,xdes);

    xcur = xdes ;
    for ider = 1:n; % for univariate iterations
        iders = iders + 1;

        search = searchi; % coordinate search direction
        search(ider) = 1.0;

        solution1 = GoldSection_nVar(funcname,tol,xcur, ...
            search,lowbound,intvl,ntrials); % 1D search
        idersall = idersall + 1;

        % if returned alpha is equal to lowerbound that
        % direction was not useful
        % reverse the direction and check
        if solution1(1) <= lowbound;
            search(ider) = -1.0;
            % reverse the search direction and
            % update x anyway
            solution1 = GoldSection_nVar(funcname,tol,xcur, ...

```

```

        search,lowbound,intvl,ntrials);           % 1D search
    for j = 3: n+2;
        xcur(j-2) = solution1(j);           %update design vector
    end
    idersall = idersall + 1;
else           % search direction acceptable ;update x
    for j = 3: n+2;
        xcur(j-2) = solution1(j);
    end
end
%store the values for iteration
xstore(iders,:) = xcur;
fstore(iders) = feval(funcname,xcur);
astore(iders) = solution1(1);
end

% begin n+1 iteration. The search is based on changes in x in the
% previous n iterations along the coordinate directions
search = xcur - xbegin;

% pause;
solution2 = GoldSection_nVar(funcname,tol,xcur, ...
    search,lowbound,intvl,ntrials);           % 1D search

if solution2(1) <= lowbound;
    search = -search;           % reverse the search direction
    solution2 = GoldSection_nVar(funcname,tol,xcur, ...
        search,lowbound,intvl,ntrials);           % 1D search
    for j = 3: n+2;           % update x even if not successful
        xcur(j-2) = solution2(j);
    end
    idersall = idersall + 1;
else
    for j = 3: n+2;           % update x
        xcur(j-2) = solution2(j);
    end
    idersall = idersall + 1;
end
iders = iders + 1;

```

```

% store the value of x
xstore(iders,:) = xcur;
fstore(iders) = feval(funcname,xcur);
astore(iders) = solution2(1);
countloop = countloop + 1;

% check convergence on f
fdiff = feval(funcname,xcur)-fbegin;
if abs(fdiff) < epsfdif;
    fprintf('Convergence in f : % 14.3E  reached in %6i cycles
\n', ...
        abs(fdiff), countloop);
    fprintf('Number of useful calls to the Golden Section Search
Method :%6i \n',iders);
    fprintf('Total number of calls to the Golden Section Search
Method :%6i \n',idersall);
    fprintf('\n The values for x and f and stepsize are \n');
    disp([xstore fstore' astore'])
    break;
end
% convergence in changes in x
xdiff = (xcur-xbegin)*(xcur-xbegin)';
if xdiff < epsxdif
    fprintf('Convergence in x : % 12.3E  reached in %6i cycles
\n', ...
        xdiff, countloop);
    fprintf('Number of useful calls to the Golden Section Search
Method :%6i \n',iders);
    fprintf('Total number of calls to the Golden Section Search
Method :%6i \n',idersall);
    fprintf('\n The values for x and f and stepsize are \n')
    disp([xstore fstore' astore'])
    break;
end
xdes = xcur; % a single iteration over
% update x and return to outloop iteration
end
if outloop == Nmax

```

```
fprintf('maximum number of cycles reached : %6i \n',Nmax);  
fprintf('Number of useful calls to the Golden Section Search  
Method :%6i \n',iders);  
fprintf('Total number of calls to the Golden Section Search  
Method :%6i \n',idersall);  
fprintf('\n The values for x and f and stepsize are \n')  
disp([xstore fstore' astore'])  
end
```

PSIM C block code

Variable/Function definitions

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

int g_nInputNodes=0;
int g_nOutputNodes=0;
int g_nStepCount=0;

/// global var definition for panel ///
double *ANS, *ANS2, *ANS3, *ANS4;

int count;
double Rs, Gsh, A, Rs_stc, Gsh_stc, A_stc;
double Io, Iph;
double gg, ff;

double y11, y22, y33, y44, y55;
double Vt;
double Isc_v, Voc_v, Impp_v, Vmpp_v, Vt_v, T_v, S_v;
double Isc_stc, Voc_stc, Impp_stc, Vmpp_stc, Vt_stc, Tc_stc, TK_stc;
double P_panel;
double v, current, p;
double temp, s;
double ss, tt;

double vol = 1.38e-23;
double q = 1.602e-19;
double Tc = 25;
double TK = 298;
double S = 1000;

//////// KC125GT //////////
double Isc = 8;
```

```

double Voc = 21.7;
double Impp = 7.2;
double Vmpp = 17.4;
double Ns = 36;
double kv = -0.0821;
double ki = 0.00318

double Example5_3(double *x)
{
    return pow(x[0]-x[1],2)+2*pow(x[1]-x[2],2)+3*pow(x[2]-1,2);
}

double Example6_1(double *x)
{
    return 3*sin(0.5+0.25*x[0]*x[1])*cos(x[0]);
}

double PVmodel(double *x)
{
    Vt = (vol*TK)/q;

    y11 = -Voc/(Ns*x[2]*Vt);
    Io = (Isc-(Voc-Isc*x[0])*x[1])*exp(y11);

    y22 = -y11;
    Iph = Io*exp(y22)+Voc*x[1];

    ////////////objective function definition//////////
    y33 = (Vmpp+Impp*x[0])/(Ns*x[2]*Vt);
    ff = Iph-Io*exp(y33)-(Vmpp+Impp*x[0])*x[1]-Impp;

    y44 = (Vmpp+Impp*x[0])/(Ns*x[2]*Vt);
    gg
    =(Io*(1/(Ns*x[2]*Vt))*exp(y44)+x[1])/(1+Io*(1/(Ns*x[2]*Vt))*
    exp(y44)+x[0]*x[1])-(Impp/Vmpp);

    return pow(ff, 2)+pow(gg, 2);
}

```

```

int i, j, cnt;
double dela;
double dx0[3],x0[3];
double dx1[3],x1[3];
double f0,f1,f1s;
double a00,a01;
int index;
double returnval[5]; // {alpha, f,x[0],x[1],x[2]}

double *UpperBound_nVar(double *x, double *s, double a0, double da,
int ns){

    for(i=1; i<ns; i++){
        j=0;          dela = j*da;          a00 = a0+dela;
        for (index=0; index<3; index++)
        {
            dx0[index] = a00*s[index];
            x0[index] = x[index]+dx0[index];
        }
        f0 = PVmodel(x0);

        j=j+1;dela= j*da;          a01 = a0+dela;
        for (index=0; index<3; index++)
        {
            dx1[index] = a01*s[index];
            x1[index] = x[index]+dx1[index];
        }
        f1 = PVmodel(x1);

        f1s = f1;
        if(f1<f0){
            for(j=2; j<ns; j++){
                a01 = a0+j*da;
                for (index=0; index<3; index++)
                {
                    dx1[index] = a01*s[index];
                    x1[index] = x[index]+dx1[index];
                }
            }
        }
    }
}

```

```

        f1 = PVmodel(x1);

        if (f1s<f1) f1s=f1;
        else      f1s=f1; // MIN(f1s,f1);
        if (f1>f1s)
        {
            returnval[0]=a01;
            returnval[1]=f1;
            returnval[2]=x1[0];
            returnval[3]=x1[1];
            returnval[4]=x1[2];
            return returnval;
        }

    }
    printf("cannot increase function in n trials");
    returnval[0]=a01;
    returnval[1]=f1;
    returnval[2]=x1[0];
    returnval[3]=x1[1];
    returnval[4]=x1[2];
    return returnval;
}
else
{
    da= 0.5*da;
}

}
returnval[0]=a0;
returnval[1]=f0;
returnval[2]=x0[0];
returnval[3]=x0[1];
returnval[4]=x0[2];
return returnval;
}

double upval[5], au, fau;
double eps1, tau,nmax;

```

```

double aL, a1, a2;
double xL[3], x_1[3], x_2[3];
double faL, fa1, fa2;

int k, h;
double returnGold[5]; // {alpha, f, x[0], x[1], x[2]}
double *AnsUpper;

double *GoldSection_nVar(double tol, double *x, double *s, double
lowbound, double intvl, int ntrials){
    AnsUpper = UpperBound_nVar(x, s, lowbound, intvl, ntrials);
    //for (cnt=0;cnt<5;cnt++)
printf("UpperBound_result[%d]=%f\n", cnt, AnsUpper[cnt]);

    for (k=0; k<5; k++)
    {
        upval[k] = AnsUpper[k];
    }

printf("upval[%d]=%f\n", cnt, AnsUpper[cnt]);

    au = upval[0];
    fau = upval[1];

    //printf("au=%f\n", upval[0]);
    //printf("fau=%f\n", upval[1]);

    eps1 = tol/(au-lowbound);
    tau = 0.38197;
    nmax = (int) floor(-2.078*log(eps1)+0.5); //반올림

    aL = lowbound;    //printf("aL_Gold = %f \n", aL);
    for (k=0; k<3; k++){
        xL[k] = x[k]+aL*s[k];
    }
    faL = PVmodel(xL);
printf("xL[%d]=%f\n", cnt, xL[cnt]);

    a1 = (1-tau)*aL + tau*au;    //printf("a1_Gold = %f \n", a1);

```

```

for (k=0; k<3; k++){
x_1[k] = x[k]+a1*s[k];
}
fa1 = PVmodel(x_1);
printf("x_1[%d]=%f\n",cnt,x_1[cnt]);

a2 = tau*aL + (1-tau)*au;    //printf("a2_Gold = %f \n", a2);
for (k=0; k<3; k++){
x_2[k] = x[k]+a2*s[k];
}
fa2 = PVmodel(x_2);
printf("x_2[%d]=%f\n",cnt,x_2[cnt]);

for(h=1; h<nmax; h++){
    if (fa1>=fa2){
        aL = a1;          faL = fa1;
        a1 = a2;          fa1 = fa2;
        a2 = tau*aL+(1-tau)*au;
        for(k=0; k<3; k++){
            x_2[k]=x[k]+a2*s[k];
        }
        fa2 = PVmodel(x_2);
    }
    else{
        au = a2;          fau = fa2;
        a2 =a1;          fa2 = fa1;
        a1 = (1-tau)*aL+tau*au;
        for(k=0; k<3; k++){
            x_1[k]=x[k]+a1*s[k];
        }
        fa1 = PVmodel(x_1);
    }
}

returnGold[0] = a1;
returnGold[1] = fa1;
returnGold[2] = x_1[0];
returnGold[3] = x_1[1];
returnGold[4] = x_1[2];

```

```

        return returnGold;
    }

    int a, aa, b, bb, n, m, mm, jj, kk, Nmax;
    double xcur[3], xcur2[3], xcur3[3], xdes[3], xbegin[3];
    double fbegin;
    double xstore[3], fstore, astore;
    double search[3], search2[3];
    double psearch[3][3]={{1,0,0},{0,1,0},{0,0,1}};
    double msearch[3][3]={{-1,0,0},{0,-1,0},{0,0,-1}};
    double fdiff, xdiff;
    double *solution1, *solution2;
    double returnPattern[5];
    double *AnsGold;
    double *PatternSearch(double tol, double *x, double lowbound, double
    intvl, int ntrials){

        for(a=0; a<3; a++){
            xdes[a] = x[a];
        }
        Nmax =100; // total pattern step

        for(n=0; n<Nmax; n++){

            for(a=0; a<3; a++){
                xbegin[a] = xdes[a];
            }
            fbegin = PVmodel(xbegin);

            for(a=0; a<3; a++){
                xcur[a] = xdes[a];
            }

            if(n>=2){ //powell shift
                for(aa=0; aa<2; aa++){
                    for(bb=0; bb<2; bb++){
                        psearch[aa][bb] = psearch[aa+1][bb];
                    }
                }
            }
        }
    }

```

```

        for(bb=0; bb<3; bb++){
            psearch[2][bb] = search2[bb];
        }
        for(aa=0; aa<3; aa++){
            for(bb=0; bb<3; bb++){
                msearch[aa][bb] = -psearch[aa][bb];
            }
        }
    }
    for(mm=0; mm<3; mm++){
        // for univariate iteration

        solution1 = GoldSection_nVar(tol, xcur, psearch[mm],
lowbound, intvl, ntrials);
        if(solution1[0] <= lowbound){
            solution1 = GoldSection_nVar(tol, xcur, msearch[mm],
lowbound, intvl, ntrials);
            xcur[0] = solution1[2];
            xcur[1] = solution1[3];
            xcur[2] = solution1[4];
        }
        else{
            xcur[0] = solution1[2];
            xcur[1] = solution1[3];
            xcur[2] = solution1[4];
        }
        fstore = PVmodel(xcur);
        astore = solution1[0];

        xcur2[mm]=xcur[mm];
    }
    printf("\n");

    search2[0] = xcur2[0] - xbegin[0];
    search2[1] = xcur2[1] - xbegin[1];
    search2[2] = xcur2[2] - xbegin[2];

    solution2 = GoldSection_nVar(tol, xcur2, search2,
lowbound, intvl, ntrials);

```

```

if(solution2[0] <= lowbound){
    search2[0] = -1*search2[0];
    search2[1] = -1*search2[1];
    search2[2] = -1*search2[2];

    solution2 = GoldSection_nVar(tol, xcur2, search2,
lowbound, intvl, ntrials);
    xcur[0] = solution2[2];
    xcur[1] = solution2[3];
    xcur[2] = solution2[4];
}
else{
    xcur[0] = solution2[2];
    xcur[1] = solution2[3];
    xcur[2] = solution2[4];
}
xstore[0] = xcur[0];
xstore[1] = xcur[1];
xstore[2] = xcur[2];
fstore = PVmodel(xcur);
astore = solution2[0];

fdiff = PVmodel(xcur)-fbegin;

if(fabs(fdiff)<1e-8){
    printf("x_result[0] = %f \n", xstore[0]);
    printf("x_result[1] = %f \n", xstore[1]);
    printf("x_result[2] = %f \n", xstore[2]);
    break;
}
xdes[0] = xcur[0];
xdes[1] = xcur[1];
xdes[2] = xcur[2];

xdiff = (xcur-xbegin)*(xcur-xbegin);
if(fabs(xdiff)<1e-8){
    printf("x_result[0] = %f \n", xstore[0]);
    printf("x_result[1] = %f \n", xstore[1]);
}

```

```
        printf("x_result[2] = %f \n", xstore[2]);  
        printf("f_result = %f \n", fstore);  
        printf("a_result = %f \n", astore);  
        break;  
    }  
}  
returnPattern[0] = astore;  
returnPattern[1] = fstore;  
returnPattern[2] = xstore[0];  
returnPattern[3] = xstore[1];  
returnPattern[4] = xstore[2];  
return returnPattern;  
}
```

OpenSimUser

```
g_nInputNodes = nInputCount;
g_nOutputNodes = nOutputCount;

double X[3] = {0, 0, 1}; // initial of Rs, Gsh, A

ANS3=PatternSearch(1e-5,X, 0, 1, 30); // (tol, x, lowbound,
incval, ntrials)

Rs = ANS3[2];
Gsh = ANS3[3];
A = ANS3[4];

Rs_stc = Rs;
Gsh_stc = Gsh;
A_stc = A;

printf("A_stc = %f \n", ANS3[4]);

TK_stc = TK;
```

RunSimUser

```

g_nStepCount++;

/// Input data for extraction ///

Isc_stc = Isc;
Voc_stc = Voc;
Impp_stc = Impp;
Vmpp_stc = Vmpp;

s = in[3];
TK = in[2] + 273;

ss = s/1000;
tt = in[2]-25;

Vt = (vol*TK)/q;

Isc = Isc_stc*ss+ki*(tt);
Voc = Voc_stc+Ns*A_stc*Vt*log(ss)+kv*(tt);
Impp = Impp_stc*ss+ki*(tt);
Vmpp = Vmpp_stc+Ns*A_stc*Vt*log(ss)+kv*(tt);

P_panel = Impp*Vmpp;
out[0] = P_panel;

printf("Voc_OPEN2 = %f \n", Voc);

double X[3] = {0, 0, 1.5}; // initial of Rs, Gsh, A
ANS4=PatternSearch(1e-5,X, 0, 1, 60); // (tol, x, lowbound,
incval, ntrials, )

v=in[0];
current=in[1];
Rs =ANS4[2];      Gsh=ANS4[3];      A=ANS4[4];

y11 = -Voc/(Ns*A*Vt);
Io = (Isc-(Voc-Isc*Rs)*Gsh)*exp(y11);

```

```
y22 = -y11;  
Iph = Io*exp(y22)+Voc*Gsh;  
  
y55=(v+ current*Rs) / (Ns*A*Vt) ;  
current = Iph - Io*(exp(y55)-1)-(v+ current*Rs)*Gsh;  
p= current*v;  
  
Isc = Isc_stc;  
Voc = Voc_stc;  
Impp = Impp_stc;  
Vmpp = Vmpp_stc;  
  
out[1] = p;  
out[2] = current;
```

Datasheet-based Modeling of Photovoltaic Panel and Its Applications

Jun-Young Park
School of Electrical Engineering,
The Graduate School,
University of Ulsan
Supervised by Prof. Sung-Jin Choi

ABSTRACT

As a simulation engine of PV simulator, PV model is becoming important in development of PV power conditioning system (PCS) for tests and evaluation of system performance. Because of the dynamic output characteristic of PV panel according to varying temperature and irradiation, it is important to replicate the PV panel and PV simulators are usually used to verify and develop the system. Therefore, an equivalent circuit model that can simulate the output characteristic of PV panel is necessary in phase of the development.

In this paper, a novel PV datasheet-based equivalent circuit model that can simulate the characteristic of PV panel is presented. The proposed model is presented to improve a drawback of the conventional modeling method. The proposed method adopts an optimization method and a dynamic datasheet update to improve the model accuracy. Verification of the proposed model compared with conventional method follows an EN50530 standard. The proposed model can be applied to PSIM model for simulation or implemented as PV simulator engine for development of PV system. Software implementation can be achieved using only PSIM program to increase the user's usability and convenience. Hardware implementation consist of the DC supply, Buck converter, and current controller. As a result, this paper presents a novel datasheet-based modeling method that can be used in both software and hardware experiments.

연구논문 실적

국제 학술지 논문 (SCI)

- [1] Jun-Young Park and Sung-Jin Choi, "A Novel Datasheet-based Parameter Extraction Method for a Single-diode Photovoltaic Array Model," *Solar Energy*, Vol. 122, pp.1235-1244, Dec., 2015.

국내 학술지 논문 (KCI)

- [1] Jun-Young Park and Sung-Jin Choi, "A New PSIM Model for PV Panels Employing Datasheet-based Parameter Tuning," *The Transactions of Korea Institute of Power Electronics (KIPE)*, Vol. 20. No. 6, Dec., 2015.

국제 학술대회 논문

- [1] Jun-Young Park and Sung-Jin Choi, "Datasheet-based circuit parameter extraction method for maximum power point simulation of photovoltaic array," *2015 ICPE-ECCE Asia 9th Conference*, pp. 1413-1419, Jun., 2015.

- [2] Jun-Young Park and Sung-Jin Choi, "Tolerance Design of Ballast resistance for Multi-string LED driver," *EPE'2015 ECCE-Europe*, pp. 1-10, Sept., 2015.

- [3] Jun-Young Park and Sung-Jin Choi, "A Novel Simulation Model for PV panels Based on Datasheet Parameter Tuning," *5th Asia-Pacific Forum on Renewable Energy (AFORE)*, Nov., 2015.

- [4] Hee-Su Choi, Jun-Young Park, and Sung-Jin Choi, "Analysis and Control of Capacitive-Coupled Wireless Power Transmission System," *15th International Conference on Control, Automation and System (ICCAS)*, Oct., 2015.

국내 학술대회 논문

- [1] 박준영, 최성진, “통계적 방법론에 기반한 선형 LED 구동회로의 최적 설계,” *전력전자 추계학술대회*, pp. 67-68, 11월, 2013.
- [2] 박준영, 최성진, “태양광 패널의 등가회로 모델링 알고리즘 개선,” *전력전자 추계학술대회*, pp. 369-370, 7월, 2014.
- [3] 박준영, 최성진, “광학링크를 이용한 전계결합형 무선전력전송 회로의 아날로그 피드백 제어,” *전력전자 추계학술대회*, pp. 3-4, 11월, 2014.
- [4] 박준영, 최성진, “최적화 방법을 이용한 MPPT 알고리즘의 개선,” *전력전자 추계학술대회*, pp. 375-376, 7월, 2015.
- [5] 최희수, 박준영, 최성진, “PSIM 모델을 이용한 변압기 회로상수 추출 방법,” *전력전자 추계학술대회*, pp. ,11월, 2015.

감사의 글

2년이라는 시간이 이렇게 금방 지나갈 지는 대학원에 입학할 당시에는 상상도 하지 못했습니다. 지금 돌이켜 보면, 석사의 2년은 상당히 짧은 시간이라고 생각합니다. 공부는 항상 뒷전이었던 제가 이 짧은 시간 내에 많은 것을 배우고 얻어갈 수 있도록 저에게 항상 적절한 조언과 동기부여를 해주시고 제가 나태해질 때는 쓴소리도 마다하지 않으신 지도교수 최성진 교수님, 연구실의 첫 졸업생으로서 언제나 후배들에게 모범이 되어야 하지만 항상 부족했던 저를 여기까지 이끌어 주신 것에 큰 감사드립니다.

바쁘신 중에도 저의 학위논문 심사를 위해서 제 부족한 논문을 읽고 아낌없는 조언을 해주신 이흥희 교수님과 전태원 교수님께도 감사드리며, 전력전자를 전공하는 학생으로서 두 분께 배웠던 전력전자와 전자회로 지식은 제가 여기까지 올 수 있도록 많은 도움을 주었습니다.

연구실에서 저와 가장 오랫동안 함께 했던 희수에게 항상 모범이 되지 못해서 미안한 마음이 들며, 앞으로 겪을 수 있는 힘든 과정을 포기하지 않고 묵묵히 앞으로 나아가라고 얘기해주고 싶습니다. 그리고 손이 부족할 때마다 옆에서 도움을 준 영태와 채호 에게도 고맙단 말과 함께 앞으로 많은 힘든 일이 있겠지만 좌절하지 않기를 당부합니다.

끝으로 아들이 큰 걱정이 없이 공부만 할 수 있도록 항상 뒤에서 묵묵히 도와주신 부모님께 언제나 감사하는 마음을 전합니다. 그 은혜를 통해서 제가 이 곳 까지 올 수 있었으며, 언제나 그 마음 잊지 않도록 하겠습니다.

지금 이 순간이 끝이 아니라 새로운 시작이라는 생각을 마지막으로 글을 마치겠습니다.

2015 년 12 월

박 준 영