

Development of Power Performance Evaluation System using Modeling Technology

설비 모델링 기술을 이용한 발전성능평가 시스템 구성방안 연구

Ji-Hoon Lee^{1†}, In-tae Lee¹, Nam-Joon Jung¹, Jung-Seok Bae¹, Young-Mo An¹
이지훈^{1†}, 이인태¹, 정남준¹, 배중석¹, 안영모¹

Abstract

Performance evaluation of a plant to efficiently manage and maintain the performance of the plant is a very important procedure. However, since the conventional performance evaluation method is an Excel-based manual method, the preparation procedure is complicated and the versatility is poor. In this paper, we analyze the problems of the existing performance evaluation system, effectively model the equipment, calculate the missing physical properties, and improve the versatility, efficiency and accuracy of the performance evaluation through the equipment modeler which performs automatic index calculation based on this.

발전소의 성능을 효율적으로 관리 및 유지하기 위한 발전소 성능평가는 매우 중요한 절차이다. 하지만 기존의 성능평가 방식은 엑셀 기반의 수작업 방식이다 보니 준비절차가 복잡하고, 범용성이 떨어지는 문제가 있었다. 이에 본 논문에서는 기존 성능평가 시스템의 문제점을 분석하고, 설비를 효과적으로 모델링하고 누락된 물성치를 계산하며, 이를 바탕으로 자동 지표계산을 수행하는 설비 모델러를 통해 성능평가의 범용성, 효율성 및 정확성을 높이는 방법을 기술하고자 한다.

Keywords: Power Plant, Plant Modeler, Power Performance Evaluation

I. 서론

발전소의 성능은 보일러, 터빈, 발전기 및 보조기기의 성능에 의해 종합적으로 평가되는 것으로서 발전소의 성능을 효율적으로 관리하고 유지하는 것은 대부분의 연료를 수입에 의존하는 우리의 현실에서는 매우 중요하다. 설비의 효율적 운영을 위한 노력이 설계, 건설, 운영 및 유지보수 시 지속적으로 이루어져야 하며, 이를 위하여 발전소의 설계가 제대로 되었는지를 평가하는 인수성능시험, 발전소의 년차 보수공사 후 발전설비의 성능이나 효율이 얼마나 개선되었는지 혹은 주요기기 교체 또는 보수 후 성능이나 효율의 개선정도를 평가하는 주기성능시험, 전력거래를 위한 각 발전설비의 효율을 평가하는 발전비용평가 성능시험이 이루어져야 한다 [1]. 이러한 성능시험을 위해서는 먼저 성능시험 결과계산 프로그램을 대상 발전소에 맞게 작성한 다음 발전소 현장에 방문하여 대상 설비의 계측기를 선정, 교정 및 설치하고, 계측기로부터

받은 일정기간의 취득데이터를 프로그램에 입력하여, 성능평가를 진행한다.

II. 국내외 기술 동향

A. 국내 기술 동향 및 수준 [1]

1) 설계기준 성능평가시스템

설계기준 성능평가 프로그램은 Version-up된 ASME Code를 기반으로 하는 Off-Line 프로그램으로서 이는 모든 조건을 발전소의 설계기준(사용연료, 기기상태, 각종 Valve 상태 등)으로 하여 발전소의 성능을 평가하게 된다. 설계기준의 성능평가 프로그램은 첫째로 발전소의 설계가 제대로 되었는지를 평가하는 인수성능시험에 활용된다. 두 번째로는 발전소의 연차 보수공사 후 발전설비의 성능이나 효율이 얼마나 개선되

Manuscript received March 14, 2017, Accepted November 27, 2018

¹ KEPCO Research Institute, Korea Electric Power Corporation, 105 Munji-ro Yuseong-gu Daejeon, 34056, Korea

† jihoon.lee@kepcoco.kr

This paper is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0>. This paper and/or Supplementary information is available at <http://journal.kepcoco.kr>.

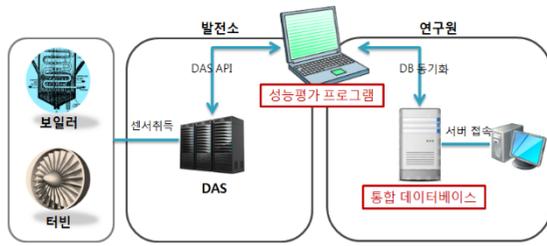


Fig. 1. 기존의 성능시험 구성도.

Description	Unit	4/4 설계	01년 인수	06년 OHA	14년 OHA
#8 FeedWater Heater					
Heating Steam Pressure	kg/cm ²	63.11	65.34	66.30	65.80
Heating Steam Temperature	°C	339.87	346.52	350.11	354.18
Heating Steam Flow	kg/hr	116,605	120,167	144,413	132,747
Heater Drain Water Temp.	°C	253.47	251.16	249.75	251.76
급수 입구온도	°C	247.91	244.46	244.83	246.76
급수 출구온도	°C	279.22	277.32	283.58	282.21
급수 유량	kg/hr	1,512,783	1,501,666	1,533,843	1,541,931
급수온도상승_TR	°C	31.31	32.87	38.75	35.45
중단온도차_TTD	°C	-1.65	2.58	-2.71	-1.83
드레인온도차_DCA	°C	5.56	6.70	4.93	5.00

Fig. 2. 기존의 성능시험 프로그램.

었는지 혹은 주요기기 교체 또는 보수 후 성능이나 효율의 개선정도를 평가하는데 활용된다. 셋째로는 전력거래를 위한 각 발전설비의 효율을 평가하는데 사용된다. 이러한 설계기준 성능평가프로그램은 전력연구원 주도하에 성능시험 절차를 만들고 이에 부합하는 Excel base 프로그램을 개발하여 운용 중에 있다. 이 프로그램은 30여 년 동안 여러사람에 의해 지속적으로 추가 및 보완이 이루어져 왔으며, 이로 인하여 프로그램 체계가 매우 복잡해졌고 실제로 Version-up된 ASME Code를 얼마나 잘 반영하고 있는지 확인하기가 어렵게 되어 있다. 그렇지만 지금까지 전력연구원을 비롯하여 한전KPS(주), 에네스지(주) 그리고 IPTC 등 국내의 모든 기관이 공통적으로 성능평가를 위하여 활용하고 있다. 최근에는 일부 외부 성능평가 기관에서 실시간 기능을 추가하고 특설계측기를 추가로 설치하여 프로그램의 기능을 보강하고자 하는 움직임이 포착되고 있다.

2) 운전상태 기준 성능평가프로그램

운전상태 성능평가 프로그램(PMS, Performance Monitoring System)은 발전소 중요기기에 대한 실시간 성능계산 및 진단을 통하여 운전원에게 최적 운전 모델을 제시하는 설비로서, 발전소 통합 정보관리 시스템에서 관련 기준 자료를 취득하여 데이터의 유효성 검증, 성능계산 및 설계 성능 대비 실적 성능 비교를 통한 공정 데이터 및 운전요소에 대한 진단 등 발전소 운영 관점에서의 실시간 성능감시시스템이 운영되고 있다.

또한 한국전력공사 전력연구원에서 발전설비의 효율적 관리와 운전성능 최적화를 위한 성능추세분석, 시간대별 및 일일 성능진단 등 실제 운전 상황을 반영한 실시간 성능감시 시스템으로 개발한 e-OASIS가 일부 화력발전소에 적용하여

Table 1. 성능감시시스템 설치현황

회사명	사업소	구축년도	회사명	사업소	구축년도
한국 서부 발전	태안 #1~4	2005	한국남동발전	분당복합	2004
	태안 #5~6	2009		여수 #2	2011
	태안 #7~8	2005	한국중부발전	신보령 #1~2	구축 중
	태안 #9~10	구축 중		판교지사	구축 중
	군산복합	2008		화성지사	
	평택화력	2011	한국지역 난방공사	과주지사	
	평택복합 #2	구축중			
서인천복합	2012				
한국 동서 발전	당진 #1~4	2006	POSCO 에너지	인천 LNG #3~4	2007
	동해 #1~2			인천 LNG #5~6	2011
	울산화력			인천 LNG #7~9	2013
	일산복합	광양 부생복합		2011	
	당진 #5~8	포항 부생복합		2013	
	호남 #1~2	2011	POSCO	포항 LNG #13~14	2005
	울산복합 #4			안양복합	2008
당진 #9~10	구축 중	GS파워	부천복합		
한국 남부 발전	하동 #1~8	2012	대우조선해양	거제(선박)	2013
	남제주#3~4		동두천파워	동두천복합	구축 중
	신인천복합				
	부산복합				
삼척 #1~2	구축 중				

* 자료출처 : 에이드(주)

운영 중에 있으며, 발전소 Trip과 관련된 사전예방정보 제공 시스템인 불시정지 예방 및 정지 원인분석 시스템(TIS, Trip-cause Information System)은 Trip이 발생되면 자동으로 원인을 추적하여 Trip 원인과 해결방법을 제공하는 시스템이 일부 발전소에 채택되어 운영되고 있다. 운전원이 발전소 중요기기에 대한 이상 상태 발생 전에 SMART Signal을 활용하여 비정상 상황으로의 변화추이를 감지하여 사고를 사전에 방지하도록 지원하는 예측 진단 및 경보시스템(PAS, Predictive Analytic System) 등 발전소 운전 및 운영측면에서의 IT를 적용한 시스템들이 운영되고 있다. 하지만 이들 감시시스템들은 IT 기술과 접목하여 감시 및 진단을 함으로써 환경 변화(상대 습도, 대기 온도 등)에 대한 보정이 이뤄지지 않으므로 발전소 운전 및 효율 측면에서 보조적인 역할을 할 수 밖에 없는 한계를 가지고 있다. 이에 따라서 운전상태의 감시를 위해서는 상기와 같이 On Line 성능감시시스템이 운영되고 있으나 보다 정밀한 주기성능진단과 인수성능시험을 위해서는 Off Line 성능평가시스템에 대한 요구가 증대되고 있으며, 이를 위해 국내 두산중공업(주), 한전KPS(주), 에네스지, IPTC 등과 같은 중대형 업체에서도 성능평가용 프로그램에 대한 독자개발을 서두르고 있는 실정이다.

B. 국외 기술 동향 및 수준 [1]

대형 발전 플랜트에 제어 및 감시용으로 적용된 분산디지털 제어설비(DCS)에서 취득한 각종 정보를 발전소 통합 정보 관리 시스템(PIMS)에서 Data를 통합 관리하고, 통합된 Data를 기반으로 플랜트에 응용 가능한 진단감시설비(성능 감시 시스템, 예측진단 및 조기경보 시스템, 발전소 불시 정지 예방 및 정지 원인분석 시스템, 발전소 건전성 감시 시스템 등)를 이용하여 효율적인 발전소 운영에 초점이 맞춰지고 있는 실정이다. 최근 국내외적으로 발전소 원격 예측진단 센터

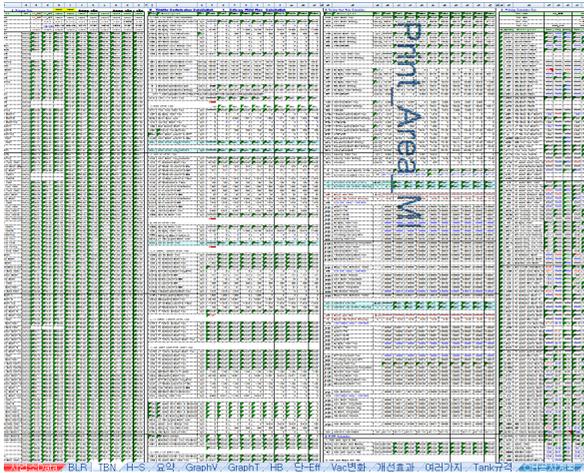


Fig. 3. 복잡한 엑셀 기반 성능시험 결과 계산 프로그램.

(Remote Predictive Diagnostic Center)를 두어 상기 진단 감시설비를 원격으로 수행하여 보다 효율적으로 운영토록 추진하고 있으나, 발전소 환경변화에 따른 근본적인 한계점을 내포하고 있을 수 밖에 없다.

III. 기존 성능시험 시스템의 문제점

기존의 성능시험 결과 계산 방식은 새로운 발전소를 대상으로 프로그램을 수정하려면, 발전설비의 구성이나 설비 별 속성 변경에 따라 관련 셀들을 일일이 찾아서 변경, 수정, 추가 및 삭제해야 한다. 이는 ASME Code와 관련된 복잡하고 연쇄적인 셀 참조들로 구성되어 있기 때문이다. 전체 프로그램을 수정하기 위해서는 수개월의 시간이 소요될 뿐만 아니라 인적 실수에 의한 평가결과의 오류 가능성도 있다.

국내에는 100개 이상의 발전소가 운영되고 있는데, 기존 엑셀 기반 프로그램은 범용성이 낮아서 다양한 발전소들의 성능평가를 수행하기 어렵다. 다양한 구성과 특징을 가진 발전소들의 성능을 평가하기 위해서 현재는 발전소 개수 만큼의 별도의 엑셀 프로그램을 운용하고 있다. 또한 발전소에서 취득되는 센서 정보 중 성능지표 계산에 필요하지만 측정되지 않는 유량이나 온도, 압력 등의 물성치 데이터에 대해서도 설계값이나 사용자 경험에 의한 추정값을 사용하고 있어 성능평가자에 따른 개인차가 발생할 수 있어 다소 정확성을 높일 필요가 있다. 본 논문에서는 기존의 발전소마다 별도로 관리되는 엑셀 방식에서 벗어나 다양한 발전소에 범용적으로 적용할 수 있고, 대상 발전소에 따른 성능평가 준비가 용이하며, 누락된 계측값을 자동으로 산출하고, 온라인으로 평가 수행이 가능할 뿐만 아니라 평가 이력에 대한 체계적인 관리가 용이한 성능평가 시스템을 제시하고자 한다.

IV. 문제 해결 방안

A. 문제 해결을 위한 요구사항

기존 시스템의 문제점을 살펴보면 공통적으로 요구되는 것은 성능평가 대상의 범용성, 성능평가 과정의 효율성, 성능

평가 결과의 정확성을 향상시키는 것이다. 이 세가지 요구사항에 대해 각각 해결방안을 도출해 보면 다음과 같다.

1) 성능평가 대상의 범용성

성능평가에 앞서 대상 발전소를 모델링해야 한다. 왜냐하면 발전소마다 설비 구성 및 속성값이 다르기 때문이다. 설비를 모델링할 때, 모든 발전소를 구성하는 단위 설비 객체를 정의하면 사용자는 아이콘 형태의 설비 객체를 드래그 앤 드롭 방식으로 UI 화면 위에 배치할 수 있다. 설비 객체들은 다양한 발전소 모델을 구성하는데 범용적으로 사용될 수 있다. 기존 엑셀 프로그램에서는 설계값, 연료분석값(Gas, Coal 등), 계측값(DAS, DCS), 성능지표계산값(ASME PTC)과 같은 심볼(Symbol)들이 객체 단위가 아닌 효율 계산 순서(보일러입력 목록, 열손실법계산, 입출력법계산, 보일러열흡수율계산 등) 또는 보고서 순서에 따라 나열되고 있다. 성능평가 대상 설비 수가 늘어나거나 설비 속성이 바뀔 때 기존 엑셀 프로그램에서는 해당 설비와 관련되는 Symbol을 엑셀 시트 곳곳에서 수작업으로 찾아서 수정해줘야 하지만, 본 논문에서는 Symbol을 소속 객체 단위로 관리하므로 대상 객체를 복제해서 갯수를 변경하거나 객체의 속성창에서 소속 Symbol을 편집하면 된다.

2) 성능평가 과정의 효율성

성능평가 시 대상 설비로부터 센서 데이터가 자동으로 취득될 필요가 있다. 모델링된 성능평가 대상 설비 객체의 각 속성에 계측기에서 제공되는 온라인 데이터를 연계하여 발전소 현장에서 들어오는 데이터를 취득한다. 모델링된 각 설비 객체마다 정의된 Symbol 중 발전소 계측값 Symbol을 발전소 데이터 취득시스템(DAS)의 Channel id와 1:1 매핑하여 온라인으로 읽어오는 것이다. 소속된 개별 Symbol의 수식을 변경하는 경우에도 기존 엑셀기반 프로그램에서는 관련된 모든 셀 참조를 확인하고 변경해 주어야 해서 작업이 어렵고 많은 시간을 필요로 하는데 비해 본 논문에서는 계산 Symbol 내의 연쇄 참조 Symbol을 자동으로 추적하고, 단계적으로 보여 주므로 Symbol 수식 변경 작업이 쉬워서 성능평가 준비작업 시간이 단축된다. 이를 위해 본 논문에서는 Symbol명이 유일성을 갖도록 정의하였다.

3) 성능평가 결과의 정확성

Symbol의 소속 객체는 보일러, 터빈, 급수가열기, 복수기 등의 설비객체와 각 설비객체를 연결하는 배관객체를 포함하는데, 객체별로 Symbol을 정의하면 객체들 간의 배관 연결 관계에 따라 증기의 흐름을 알 수 있으므로 누락된 유량 및 물성치(온도, 압력, 엔탈피, 엔트로피, 비체적, 건도)에 대한 자동 계산이 가능하다. 발전소 모델의 각 객체에 필요한 유량 정보 중 DAS로부터 취득되지 않는 값은 유량방정식 계산을 통해 구한다. 유량방정식은 각 설비 객체의 특성과 질량 보존 법칙에 의해 자동으로 생성한다. 엔탈피를 비롯한 물성치값은 각 설비 객체의 특성과 에너지 보존 법칙에 의해 생성한 에너지 방정식을 통해 자동으로 구한다. 유량 및 물성치 계산이 완료되면, 정의된 Symbol 중 성능지표 Symbol에 대한 계산을 수행할 수 있다. 보일러, 터빈 각각의 성능지표 Symbol은 Symbol 간에 다단계의 복잡한 참조 연산을 통해 계산되는데, 최종 참조값은 설계값, 연료분석값, 계측값과 같이 수치로 된

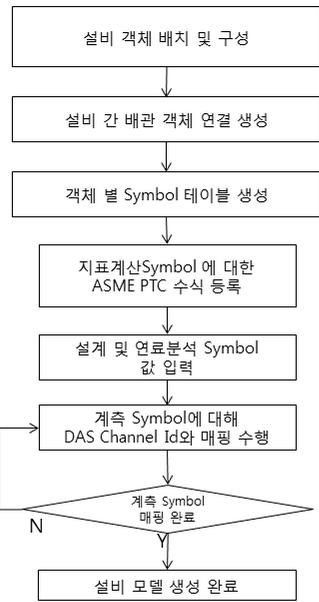


Fig. 4. 설비 구성 순서도.

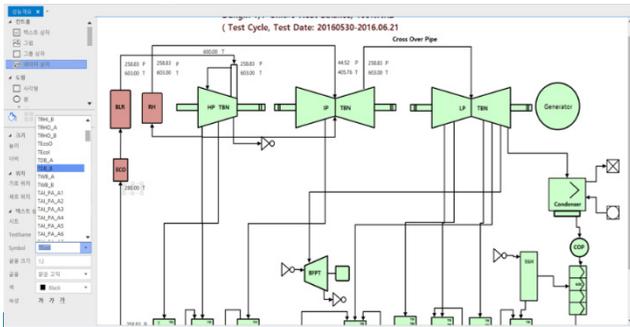


Fig. 5. 설비 모델러 화면.

Symbol을 참조한다. 기존의 성능시험 결과계산 프로그램은 이러한 참조의 흐름이 엑셀시트의 셀 참조로 복잡하게 얽혀 있어 전체 수식의 계산 흐름을 추적하거나 값을 검증하기 어려운 문제점이 있다. 본 논문에서는 모델링 과정에서 설비 객체들 간의 연결관계 및 각 객체별로 정의된 Symbol을 기반으로 성능지표 계산식을 추출 및 생성할 수 있고, 자동으로 반복계산을 수행할 수 있다. 위에서 설명한 내용을 종합하면 성능평가의 범용성과 효율성, 그리고 정확성을 만족시키기 위해서는 성능평가의 대상이 되는 발전소 설비들을 단위 설비 객체로 정의하고, 이 객체들을 다양한 방식으로 조합하고 연결하여 발전소를 모델링할 수 있으며, 각 객체별로 정의된 Symbol들에 대해 실시간으로 데이터를 취득 또는 계산할 수 있는 시스템이 필요한데, 이를 설비 모델러라고 정의한다.

B. 설비 모델러 구현방안

1) 설비 객체 구성

먼저 보일러, 터빈, 복수기, 펌프, 급수가열기 등과 같은 발전소 설비 객체들을 드래그 앤 드랍 방식으로 배치하고, 설비 객체들 간에 배관 연결을 생성한다.

각 객체별로 Symbol 테이블을 등록하는데, Symbol 테이블을 등록할 때는 설계값, 연료분석값(Gas, Coal등), 계측값(DAS,DCS), 성능지표계산(열손실법계산, 입출력법계산, 보일

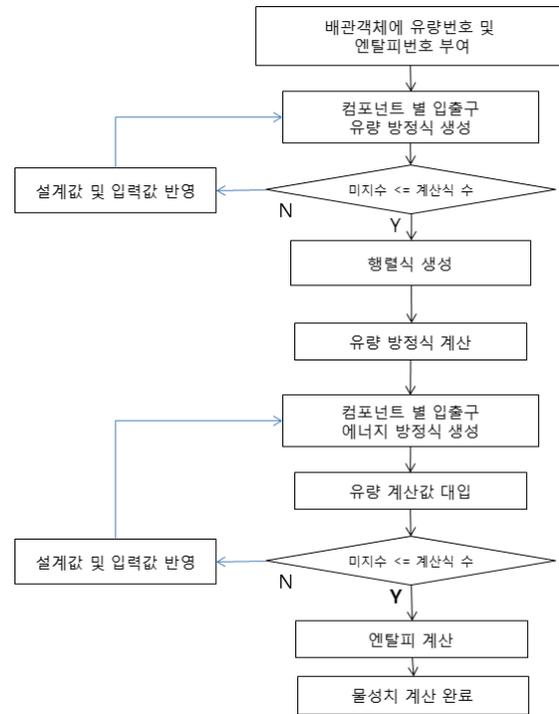


Fig. 6. 물성치 계산 순서도.

리열흡수율계산 등)와 같은 그룹 별로 탭을 구분하여 등록한다. 이는 기존 성능평가 프로그램에 호환성을 갖기 위함임. 이를 통해 기존의 엑셀 시트와 유사한 형식으로 Symbol 수식 및 값을 연계할 수 있다. Symbol 중 성능지표 Symbol은 ASME PTC기반의 성능계산 수식을 사용자가 직접 입력할 수도 있고, 이미 만들어진 타 발전소 모델이 있는 경우, 이 참조 모델의 수식을 읽어서 변경 사용할 수 있다. 설계값 Symbol과 연료분석값 Symbol은 별도로 제공되는 설계, 분석파일을 읽어서 적용하는 것이 기본이고, 성능지표 Symbol처럼 참조 모델이나 직접 입력 방식으로도 사용할 수 있다. 생성된 Symbol 테이블 중 계측 Symbol은 데이터 취득장치의 Channel id와의 매핑 작업을 수행함. 이때, Channel id는 DAS로부터 읽어온 id 목록 중에서 선택 가능함. Symbol과 Channel 정보 매핑이 완료되면 설비 모델 생성이 완료된다.

2) 물성치 계산

설비 객체 구성 단계에서 계측 Symbol 및 Channel id 매핑작업 이후에도 값이 취득되지 않는 계측 Symbol이 있다. 데이터가 취득되지 않는 계측 Symbol은 유량 및 물성치에 해당되는 값인데, 이를 구하기 위한 절차는 Fig. 6과 같다.

설비 객체 구성 단계에서 생성한 각 배관 객체에 Fig. 7과 같이 번호를 부여한다.

설비 객체 별 입출력 특성을 고려하여 앞서 배관 객체에 부여된 번호로 유량변수(M)를 정의하여 유량방정식을 생성한다. Table 2는 설비 객체 별 입출력 포트 및 유량방정식 생성의 예이다.

$$\text{Standard Outlet} = \text{Standard Inlet} + \text{Extraction.}$$

이 식에 유량번호를 적용한 유량방정식은 $M16 = M12 + M17$ 이다.

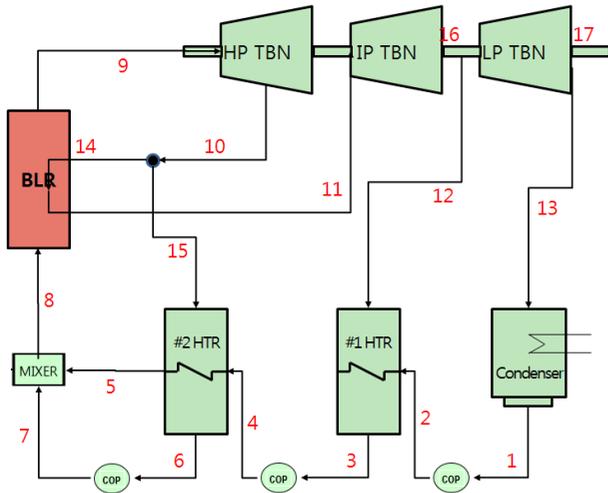


Fig. 7. 설비 모델러(200) 물성치 계산 화면.

Table 2. 터빈 입출력[3]

구 조		포트 ID	설 명
		I	Standard Inlet
	O	Standard Outlet	
	E	Extraction	

Table 3. 보일러 입출력[3]

구 조		포트 ID	설 명
		IA*	Primary Inlet
	IB*	Secondary Inlet	
	O	Mixed Outlet	

Mixed Outlet = Primary Inlet + Secondary Inlet.

이 식에 유량번호를 적용한 유량방정식은 $M8 = M5 + M7$ 이다.

전체 설비 객체 유량방정식에서 미지수가 방정식 개수보다 크면 결과 계산이 어려우므로 설계치를 대입하거나 사용자가 추가로 입력하여 미지수가 방정식 개수보다 같거나 작도록 수정한다. 각 유량방정식을 행렬로 재구성한다. 행렬 연산은 프로그램으로 구현하기 용이한 방법인 가우스 소거법을 사용한다. 유량방정식 행렬 계산 수행 절차는 아래와 같다.

- 행렬을 구성한다. (유량번호 3,4,5,8번의 유량값을 알고 있다고 가정)
- 대각원소 이하 하삼각 행렬이 모두 0이 되도록 행 자리바꿈 및 피벗연산을 수행한다. (여기서 피벗이란 소거의 기준이 되는 대각원소를 말한다.) 행 자리바꿈은 대상 행의 대각원소가 0인 경우 이 대각원소와 같은 열의 원소가 0이 아닌 다른 행과 자리바꿈하는 것이다.
- 자리바꿈 후에도 대각원소 아래 하삼각행렬에 0이 아닌 값이 존재하므로 이를 소거하기 위해 피벗연산을 수행한다. Fig. 9의 수식에서 m_{i1} 은 i 행 1열의 승수, a_{i1} 은 i 행 1열의 원소, c_i 는 i 행의 상수를 의미한다. Fig. 9에서 1열의 승수(multiplier)인 m_{i1} 은 피벗원소인 a_{11} 와 나머지 각 행의 1열 원소를 이용해서 구해진다. 이 승수를 1행의 각

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} M1 \\ M2 \\ M6 \\ M7 \\ M9 \\ M10 \\ M11 \\ M12 \\ M13 \\ M14 \\ M15 \\ M16 \\ M17 \end{pmatrix} = \begin{pmatrix} 0 \\ 100 \\ 0 \\ 0 \\ 0 \\ -100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Fig. 8. 유량계산 행렬식.

$$m_{i1} = \frac{a_{i1}}{a_{11}} \quad (i = 2, \dots, n)$$

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)} \quad (j = 2, \dots, n)$$

$$c_i^{(2)} = c_i^{(1)} - m_{i1}c_1^{(1)}$$

Fig. 9. 피벗(Pivot) 연산식.

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} M1 \\ M2 \\ M6 \\ M7 \\ M9 \\ M10 \\ M11 \\ M12 \\ M13 \\ M14 \\ M15 \\ M16 \\ M17 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -100 \\ -100 \\ 0 \\ 0 \\ 100 \\ 0 \\ -100 \\ -100 \\ 0 \\ 0 \\ 100 \\ -100 \\ -100 \\ 0 \end{pmatrix}$$

Fig. 10. 유량계산 행렬식 계산 결과.

열에 곱한 값을 i 행의 각 열에서 빼주면 i 행의 새로운 원소값이 된다. 2열에서 n 열까지도 같은 방법으로 계산을 수행하면 대각원소 하단의 원소들은 Fig. 10과 같이 모두 0으로 소거된다.

- 피벗 연산 완료 후 대각원소 중 값이 없는 부분이 존재하면 결과 계산이 어려우므로 해당 유량값은 설계치를 대입하거나 사용자가 추가로 입력하여 대각원소가 존재하도록 수정한다.
- 마지막으로 후진대입법을 이용하여 각 원소 즉, 유량 Symbol의 값을 구한다. 후진대입법은 먼저, 마지막 행의 유량방정식을 통해 유량 변수값을 구하고, 이 값을 바로 상위 행에 대입하여 그 행의 유량 변수값을 구하는 방식으로 이를 반복적으로 수행하여 전체 유량변수를 구한다. 설비 객체 별 입출력 특성을 고려하여 유량변수(M), 엔탈피변수(E), 일변수(W), 열량변수(Q)로 에너지방정식을 생성한다. Table 4는 에너지방정식 생성의 예이다.

$$\text{Mass Inlet} \times \text{Enthalpy Inlet} + W = \text{Mass Outlet} \times \text{Enthalpy Outlet}.$$

이 식을 배관번호로 표현한 에너지방정식은 $M1 \times E1 + W1 = M2 \times E2$ 이다.

Table 4. 펌프 입출력 [3]

구조		포트 ID	설명
		I	Standard Inlet
		O	Standard Outlet

Table 5. 보일러 입출력 [3]

구조		포트 ID	설명
		IT	Tube Inlet
		OT	Tube Outlet

Table 6. 터빈 입출력 [3]

구조		포트 ID	설명
		I	Standard Inlet
		O	Standard Outlet
		E	Extraction

Table 7. 그룹별 Symbol 테이블

5) IP Turbine Inlet Steam Flow				
MHRS	Hot Rehear Steam Flow	kg/hr	0	MHRS
MLK50	HP TBN to IP TBN Cooling Steam Flow	kg/hr	0	MLK50
MIP1	IP Turbine inlet Steam Flow	kg/hr	0	MHRS+MLK50

Mass Inlet × Enthalpy Inlet + Q = Mass Outlet × Enthalpy Outlet.

이 식을 배관번호로 표현한 에너지방정식은 M8×E8+Q8 = M9×E9이다.

Mass Inlet × Enthalpy Inlet = Mass Outlet × Enthalpy Outlet + Mass Extraction × Enthalpy Extraction.

이 식의 에너지방정식은 M11×E11 = M16×E16+W1 이다.

유량방정식 계산에서 구해진 유량변수 값들을 에너지 방정식에 대입하여 유량변수를 소거한다. 전체 설비 객체 에너지방정식에서 미지수가 방정식 개수보다 크면 결과 계산이 어려우므로 설계치를 대입하거나 사용자가 추가로 입력하도록 한다. E, Q, W를 구하는 과정은 S225의 행렬계산 과정과 동일하게 가우스 소거법을 수행한다.

3) 성능지표 자동 계산

설비 객체 구성 단계에서는 기존 성능평가 프로그램의 엑셀 방식에 호환성을 제공하기 위해 그룹별로 Symbol을 등록하였다. 이를 바탕으로 성능지표 계산을 자동으로 수행하는 절차에 대해 설명한다.

설비 객체 구성 단계에서 각 객체별 Symbol 테이블 그룹을 엑셀 시트에서 검색하여 없으면 새로 생성한다. 새로 생성할 때는 그룹 번호 순서대로 생성한다. 그룹에 속하는 연료 분석Symbol, 성능지표Symbol, 계측Symbol, 설계Symbol 등의 각 Symbol들은 엑셀 시트에 생성할 때, 각기 해당하는 그룹명 아래에 생성한다. Table 7은 IP 터빈 객체에 등록된 『5) IP Turbine inlet Steam Flow』라는 그룹에 등록된 Symbol들을 엑셀 시트에 생성한 예이다.

그룹별 Symbol 테이블 갱신 완료시까지 반복 수행한다.

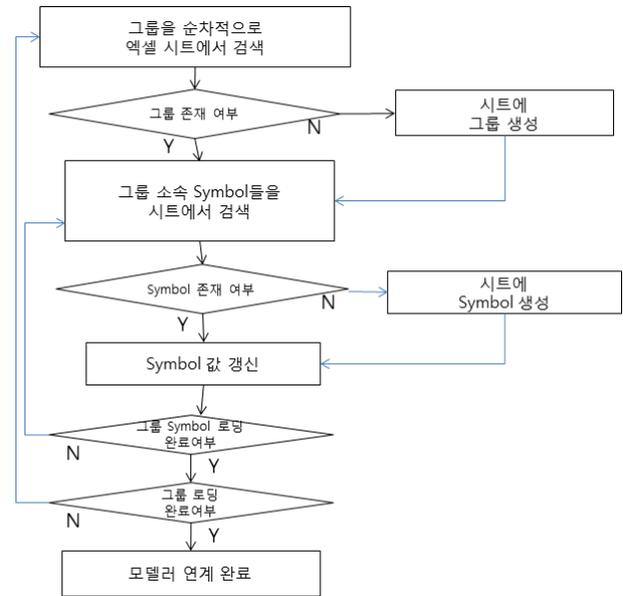


Fig. 11. 설비 모델 연계 순서도.

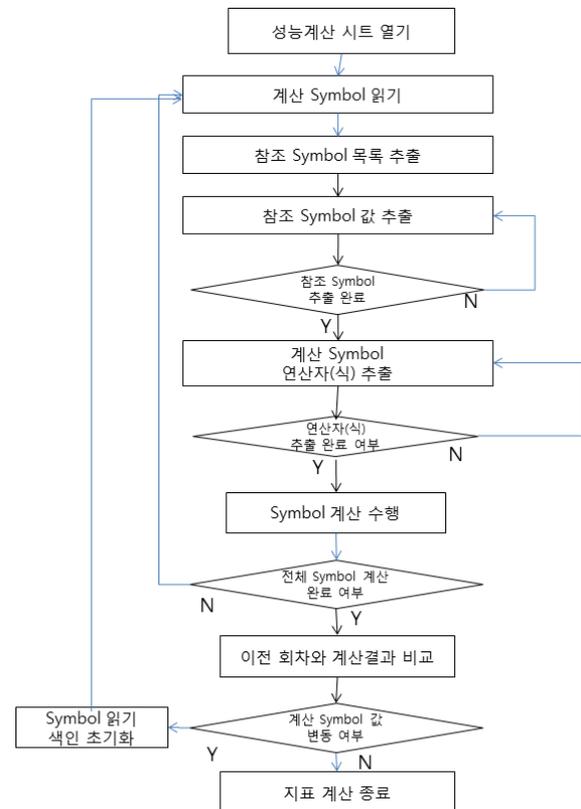


Fig. 12. 성능지표 계산 순서도.

여기서 기존 성능평가 프로그램의 엑셀 시트와의 차이점은 첫째, 설비 모델로부터 자동생성 된다는 것과 둘째, Symbol의 값 부분에 셀 참조 대신 계산 수식이 들어간다는 것이다. 이 계산 수식을 활용하여 성능지표 자동 계산을 수행한다.

모델러 연결부(310)에서 생성한 엑셀 시트를 읽는다. 성능지표Symbol의 수식에는 다수개의 참조Symbol들이 들어 있고, 각 참조 Symbol들은 하부에 연쇄 참조 구조를 가지고 있다. 이를 풀기 위해 먼저, 성능지표Symbol의 수식에 포함된 참조Symbol들에 대한 목록을 추출하여 배열을 생성한다.

6. ELEP Calculation				
1) Total Heat Input to Turbine				
QMS_e	Main Steam Heat Quantity	Mcal/h	0	MMS * HMS /1000
QHRS	Reheat Steam Heat Quantity	Mcal/h	0	MRS * HRS / 1000
Qin	Total Heat Input to Turbine	Mcal/h	0	GMS + QHRS
2) Total Heat Output from Turbine				
Q6E	#6 Extraction Steam Heat	Mcal/h	0	TBNF.M6E * TBND.H6E / 1000
Q7E	#7 Extraction Steam Heat	Mcal/h	0	TBNF.M7E * HCR / 1000
Q6E	#6 Extraction Steam Heat	Mcal/h	0	TBNF.M6E * TBND.H6E / 1000
Q5E	#5 Extraction Steam Involve BFPT	Mcal/h	0	(TBNF.M5E+TBNF.M5BFP)*TBND.H5E/1000
Q4E	#4 Extraction Steam Heat	Mcal/h	0	TBNF.M4E*TBND.H4E/1000
Q3E	#3 Extraction Steam Heat	Mcal/h	0	TBNF.M3E*TBND.H3E/1000
Q2E	#2 Extraction Steam Heat	Mcal/h	0	TBNF.M2E * TBND.H2E / 1000
Q1E	#1 Extraction Steam Heat	Mcal/h	0	TBNF.M1E * TBND.H1E / 1000
QCRS	Cold Reheat Steam Heat	Mcal/h	0	TBNF.MCRS * HCR / 1000
QLK2	CV Stem 3rd Leak-off Steam to SSH	Mcal/h	0	TBND.MLK2 * HMS / 1000
QLK40	CV Stem 2nd Leak-off Steam to #6	Mcal/h	0	TBND.MLK40 * HMS / 1000

Fig. 13. 성능지표Symbol 내 참조Symbol.

```
public static int CellCalc(CalcDataGrid CurGrid, int row, int col, bool calc)
{
    string Tip = CurGrid.Rows[row].Cells[col].ToolTipText; // 참조Symbol 읽어오기
    if(Tip=="") return -2; // 값 없으면 리턴
    if(CurGrid.Rows[row].Cells[col].Style.BackColor==Color.LightGreen)return 1; //계산완료면 리턴
    string[] symArray = SymbolInString(Tip).Split(','); // 참조Symbol 목록 추출

    for(int i=0; i<symArray.Length; i++)
    {
        CalcResult resultV = getSheetValue(CurGrid, symArray[i], col); // 각 참조Symbol 값 추출
        Tip = Regex.Replace(Tip, symArray[i], resultV.value); // 참조Symbol 자리에 값 대입하여
        // 변경된 수식 생성
    }
}
```

Fig. 14. 참조Symbol 배열 생성 pseudo 코드.

```
ELEP : UEPP-0.87*ELX*(1-0.01*Y)*(1-0.0065*Y) :
ueep : Qjo / MLPE * 1000 :
ebx : 0.555556*(15.2-(15.2-19.8)/(800-900))*(800-VAN_f) :
y : (1-(ELEP-'h')/(h-'h'))*100 :
y : (1-(ELEP-'h')/(h-'h'))*100 :
```

Fig. 16. Symbol에 포함된 연산식의 예.

```
public string Calculate(string text)
{
    string backup = text;
    Expression ex = new Expression(text); // 계산식 분석 클래스
    ex.EvaluateFunction += delegate(string name, FunctionArgs args) // 계산 규칙 정의
    {
        double sum=0;
        if(name=="sum") // 덧셈 계산 구현
        {
            for(int i=0; i<args.Parameters.Length; i++)
                sum += Convert.ToDouble(args.Parameters[i].Evaluate());
            args.Result = sum;
        }
        else if(name=="pmt_seaempct") // PMT Util 계산 구현
        {
            double arg1 = Convert.ToDouble(args.Parameters[0].Evaluate());
            args.Result = PMT_SeaTempCT(arg1);
        }
        else // Steam Table 계산 구현
        {
            MethodInfo mInfo = typeof(SteamTable).GetMethod("part"+name);
            if(mInfo==null) return;
            ParameterInfo[] parameters = mInfo.GetParameters();
            SteamTable stmTable = new SteamTable();
            if(parameters.Length==3)
            {
                object[] param = new object[parameters.Length];
                param[0] = Convert.ToDouble(args.Parameters[0].Evaluate()); // 파라미터#1
                param[1] = Convert.ToDouble(args.Parameters[1].Evaluate()); // 파라미터#2
                param[2] = Convert.ToInt16(args.Parameters[2].Evaluate()); // 파라미터#3
                args.Result = (double)mInfo.Invoke(stmTable, param);
            }
        }
    };
    return text; // 계산결과 반환
}
```

Fig. 17. Symbol 계산 pseudo 코드.

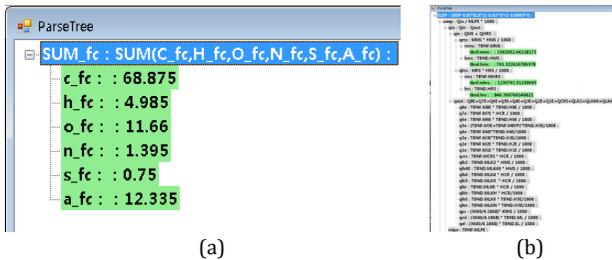


Fig. 15. (a) 연료분석Symbol, (b) 성능지표Symbol.

모든 Symbol은 고유한 Symbol명을 갖고 있으므로 메모리에 로딩된 Symbol들 중 필요한 참조Symbol들을 추적할 수 있다. 추적된 참조Symbol이 계측Symbol, 설계Symbol 또는 일부 연료분석Symbol처럼 수치값을 가진 Symbol이라면 그 값을 반환하고, 그게 아니라 또다른 참조Symbol로 이루어져 있다면 0을 반환한다.

Symbol에 포함된 연산자 또는 연산식을 추출한다.

연산식 중에는 일반적인 연산식(sum, sub 등) 이외에 외부 연산식이 있다. 외부 연산식은 pmt util, WinSteam과 같은 열효율 계산에 필요한 외부 라이브러리 연산식이다. 추출된 연산자(연산식)에 참조Symbol에서 구해진 값을 대입하여 만들어진 수식의 계산을 진행하면, 미리 정의해 놓은 연산자(연산식) 규칙에 따라 수식 계산을 수행한다. 이런 방식으로 메모리 상의 모든 Symbol들에 대한 계산을 수행한다. Fig. 16의 코드는 연산식을 처리할 수 있도록 계산 규칙을 반영한 예다.

Symbol의 참조 깊이는 1~8단계 정도이므로 S323의 과정을 한번 수행해서는 최종 값을 구할 수 없다. 참조Symbol의 깊이 만큼의 반복 계산이 필요하다. 하지만, 메모리에 로딩된 Symbol들의 깊이는 모두 제각각이다.

Table 8. Symbol 참조 Level [5]

LEVEL	1	2	3	4	5	6	7	변수번호	SYMBOL
	1							3.2.5.1	MDV
6) Drum Level Change Flow(3.3.6)									
	1	2	3	4	5	6		2.3.6.17	IMDM
								2.3.6.5	V
								2.3.6.12	V1
								2.3.6.11	COS_A1
								2.3.6.10	A1
								참조	X1
								참조	A1

각 Symbol을 재귀적으로 추적하는 방식으로 계산할 수도 있지만, 재귀호출은 Symbol들 간에 참조Symbol을 중복계산할 수 있고 검색시간이 오래 걸리므로 비효율적이다. 본 논문에서의 지표계산 방식은 아래와 같이 반복 계산하는 과정에서 점진적으로 최고Level Symbol까지 계산이 완료된다.

- 첫 번째 Symbol부터 시작해서 순차적으로 Symbol 계산을 수행한다.
- 마지막 Symbol까지 계산이 완료된 후 계산 결과를 저장하고, 다시 처음 Symbol로 가서 순차적으로 계산을 수행한다.
- 이 때, 이전 회차의 계산 결과와 계산값(수식)이 달라진 Symbol이 있는지 체크하면서 계산을 수행한다. Symbol 계산 중 값이 구해져서 더 이상 변하지 않는 Symbol은 계산 대상에서 제외한다. Symbol 최고 Level 만큼의 반복 수행이 필요하다. (약 8 회)
- 일반적인 계산 이외에 비열계산, 터빈유량계산, ELEP(저압터빈 종단 엔탈피)계산, 그룹 1 보정과 같이 예측값과

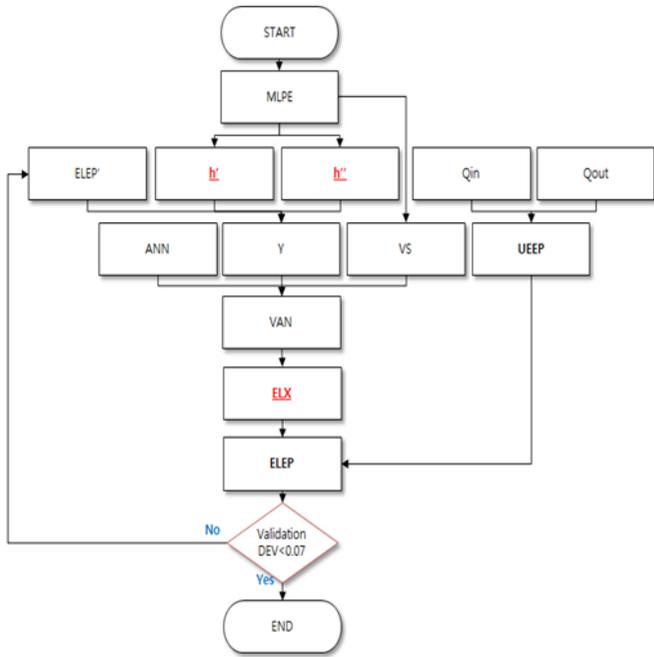


Fig. 18. Symbol 반복 계산 흐름도 [2].

Table 9. Symbol 반복 계산 예시

Symbol	Description	Unit	Value	Symbol	Description	Unit	Value
Qin_Out	LP Turbine Exhaust Expansion Line End Point		887.687	Qin - Qout + QLK LPC		888.912	888.912
MLPE	LP Turbine Heat to Condenser	Mcal/h	1,575.432	MLPE		1,581.275	1,581.275
UEEP	LP Turbine Exhaust Steam Flow	kg/h	562.029	Qin_Out*1000 / MLPE		562.149	562.149
ELEP	LPE Used Energy End Point	kcal/kg	557.590	ELEP = (1 - PWR) * (EELP - EELP) / (1 - PWR)		557.708	557.708
PLPE	LP Turbine Exhaust Steam Pressure	kg/cm ²	0.05180	ENT PHW		0.05180	0.05180
h'	LPE Saturated Water Enthalpy	kcal/kg	33.19	StmPQH(PLPE 0.6)		33.19	33.19
h''	LPE Saturated Steam Enthalpy	kcal/kg	611.75	StmPQH(PLPE 1.6)		611.75	611.75
Y	LPE Steam Moisture	%	9.36	(1 - (EELP - h' - h'')) * 100		9.34	9.34
VS	LPE Saturated Steam Specific Volume	m ³ /kg	27.77	StmPQH(PLPE 1.6)		27.77	27.77
ANN	LPE Stage Annulus Area	m ²	54.00	ANN*1		54.80	54.80
VAN_ms	LPE Stage Annulus Velocity	m/sec	201.51	MLPE*(1-Y)/100*(VS/(3600*ANN))		201.79	201.79
VAN_fs	LPE Stage Annulus Velocity	f/sec	661.130	VAN_ms/0.3048		662.050	662.050
ELX	LP Turbine Exhaust Loss at Curve	kcal/kg	5.9944	1 / (1.8 * 10.79)		5.9944	5.9944
ELEP	LPE Expansion Line End Point	kcal/kg	557.590	UEEP - EELP * ELX * (1 - 0.91 * Y) * (1 - 0.0001 * Y)		557.708	557.708

실제값이 수렴해야 하는 경우에도 위의 반복계산 방법이 유효하다. 왜냐하면 수렴이 완료되기 전까지는 상호참조된 예측Symbol과 실제Symbol의 값이 계속 변하기 때문이다.

Table 9는 ELEP(저압터빈 종단 엔탈피 계산)를 구하는 계산의 예이다. ELEP, ELEP'는 상호참조하며 ELEP와 ELEP'의 차가 오차범위(0.07)내로 수렴한 후에는 두 Symbol 모두 값이 변하지 않는다. 반대로 오차범위 내로 수렴하기 전까지는 값이 계속 변동되므로 화살표 방향으로 반복 계산을 하게 된다. 참고로 ELEP 계산을 순서대로 표시하면 Fig. 18과 같다. 성능지표계산 결과 화면의 예시는 Fig. 19와 같다. Fig. 20과 같이 발전소 설비별 주요 효율 지표를 모니터링할 수 있다.

V. 결론

본 논문에서는 기존에 발전소마다 별도로 관리해오던 성능평가 프로그램을 하나의 범용 시스템으로 통합하였다. 설비 모델러를 통해서 다양한 발전소를 모델링 할 수 있고, 이를 기반으로 누락된 센서데이터를 자동으로 계산하여 기존 방식에 비해 정확한 성능계산이 가능하다. Symbol 내의 연쇄 참조

Fig. 19. 성능지표 계산결과.



Fig. 20. 효율 지표 화면.

Symbol들을 자동 추적 및 관리할 수 있으므로 기존 엑셀 기반 성능평가 프로그램 대비 효율적인 성능평가 준비 작업을 수행할 수 있고, DAS와 온라인으로 연결하여 취득한 데이터를 바탕으로 효율적인 성능계산이 가능하다. 국내에는 120여기의 발전소에 대해 100여건의 성능평가가 수행되고, 국외에는 주요 12개국에 45,000여건의 성능평가 시장이 형성되어 있다. 본 논문의 성능평가 방법을 통해 성능평가의 기술 경쟁력을 향상하면 연간 약 22,500억원 시장에서 높은 수익성을 기대할 수 있을 것으로 예상된다.[1] 현재 본 연구는 프로그램 설계 및 개발 중에 있으며 향후에는 본 시스템의 현장 테스트를 통해 다양한 상황에 대한 신뢰성 만족시킬 수 있도록 검증을 진행할 예정이다.

REFERENCES

- [1] 안연식, "발전설비 On-Site 성능평가프로그램 개발", 연구개발사업계획서, 2014, pp7-12.
- [2] 이인태, "연구개발 수행내용 및 결과", 성능평가 현장실태조사, 2016, pp14.
- [3] 이인태, "발전설비 성능평가 시스템 [모델러]", SW구현보고서, 2016, pp11-19
- [4] 흥은기, "DJTP4_PT_OHA-T_1407", 성능평가 프로그램, 2014.
- [5] 송덕용, "발전소 화력 성능평가 알고리즘_V2R0", 2016.
- [6] 김성돈, "PowerCal용 당진4 화력성능지표", 2016.
- [7] 변화은, "대형 화력발전플랜트 통합 정보관리 및 진단 감시 시스템 적용기술 동향", 2010.