

建築研究資料

Building Research Data

No. 190

April 2019

各種空調設備システムの 潜熱負荷処理メカニズムを踏まえた エネルギー消費量評価法に関する検討

A Study on Energy Consumption Evaluation Method Considering Latent Heat
Load Processing Mechanism of Air Conditioning Systems

三浦尚志、羽原宏美、荻野登司、宮田征門、小椋大輔、巽佑介、
宮島賢一、峰野悟、松井伸樹、西村忠文、佐藤大輔

Hisashi MIURA, Hiromi HABARA, Takashi OGINO, Masato MIYATA,
Daisuke OGURA, Yusuke TATSUMI, Kenichi MIYAJIMA, Satoru MINENO,
Nobuki MATSUI, Tadahumi NISHIMURA, Daisuke SATO

国立研究開発法人 建築研究所

Published by

Building Research Institute

National Research and Development Agency, Japan

はしがき

喫緊の課題である建築物の省エネルギー化を達成するためには、建築物を設計した時点でエネルギー消費量を評価し、より適切な設計に誘導していくことが重要である。これを実現することを目指して、建築研究所では、個別研究開発課題として「省エネ基準運用強化に向けた住宅・建築の省エネルギー性能評価手法の高度化(平成 23～25 年度)」及び「建築物の省エネ基準運用強化に向けた性能評価手法の検証および体系化(平成 26～27 年度)」を行い、住宅・建築物の省エネルギー基準に適用可能な一次エネルギー消費量を指標とする総合的評価方法の開発を行ってきた。しかし、現状では、暖冷房負荷計算時の湿度および潜熱負荷を正確に把握することができず、非住宅建築物で近年導入が増えつつある潜顕分離空調システムについては十分に評価されているとは言い難い。また、住宅においても、在室者が冷房を行うかどうかの判断には、湿度の多寡が大きく影響を与えていることは自明であるが、居住者の冷房に関する行為や除湿の制御方法等に関する知見が十分でないため、除湿制御を組み込んだ機器(エアコンの除湿機能等)や全熱交換器の湿気調節効果、窓開閉による自然風利用が十分に評価されているとは言い難い。

これらの課題を解決するために、国土交通省建築基準整備促進事業の調査事項「E3 各種空調設備システムの潜熱負荷処理メカニズムを踏まえたエネルギー消費量評価法に関する検討(平成 25～27 年度)」の事業主体(株式会社建築環境ソリューションズ、ダイキン工業株式会社)との共同研究を実施した。建築研究所はこの共同研究で、調査研究の計画策定、研究成果の取りまとめに関して主たる役割を果たした他、個別の実験・調査等についても事業主体とともにを行い、最終的に潜熱負荷処理メカニズムを踏まえたエネルギー消費量評価手法を提案している。

本資料は、上記の建築基準整備促進事業「E3 各種空調設備システムの潜熱負荷処理メカニズムを踏まえたエネルギー消費量評価法に関する検討」において、平成 25～27 年度の3カ年で実施した調査報告書を再構成して、3年間の調査内容全体を取りまとめたものである。本資料では、潜熱負荷に関わる基礎的な実験データや、一次エネルギー消費量の評価法に潜熱処理の影響を反映するための方法等が詳細に整理されている。本資料が、省エネルギー基準をよりよく理解するうえで、また、省エネルギー性能の優れた住宅・建築物を設計するうえで役立てられることを期待する。

平成 31 年 4 月
国立研究開発法人建築研究所
理事長 緑川 光正

各種空調設備システムの潜熱負荷処理メカニズムを踏まえた

エネルギー消費量評価法に関する検討

三浦尚志^{*1}、羽原宏美^{*1}、荻野登司^{*1}、宮田征門^{*2}、
小椋大輔^{*3}、巽佑介^{*3}、宮島賢一^{*4}、峰野悟^{*4}、
松井伸樹^{*5}、西村忠文^{*5}、佐藤大輔^{*5}

概 要

本調査の目的は、住宅及び建築物の省エネルギー基準における一次エネルギー消費量の評価法に潜熱処理の影響を反映するための方法を開発することである。まず、住宅、非住宅ともに、生活行動パターンや建築特性、潜熱負荷発生条件に関する知見を収集し、これを元に潜熱負荷を適切に考慮した熱負荷計算法の構築を行い、潜熱処理の影響を考慮した負荷計算方法を開発した。また、デシカントシステム、家庭用エアコン（除湿モード）を対象とした実験室実験を行い、建物側の負荷特性や外気条件ごとの吸放湿特性を明確にした。また吸放湿特性を再現する数理モデルの構築を行った。これらの結果を基に、潜熱負荷処理メカニズムを踏まえた空調設備システム評価法の体系を明らかにし、省エネ基準の評価ロジックに反映するための方法を示した。

国立研究開発法人建築研究所^{*1}、国土技術政策総合研究所^{*2}、
国立大学法人京都大学^{*3}、株式会社建築環境ソリューションズ^{*4}、
ダイキン工業株式会社^{*5}

※ 所属は研究終了時点（平成 28 年 3 月）

A Study on Energy Consumption Evaluation Method Considering Latent Heat Load Processing Mechanism of Air Conditioning Systems

Hisashi MIURA^{*1}, Hiromi HABARA^{*1}, Takashi OGINO^{*1}, Masato MIYATA^{*2},
Daisuke OGURA^{*3}, Yusuke TATSUMI^{*3},
Kenichi MIYAJIMA^{*4}, Satoru MINENO^{*4},
Nobuki MATSUI^{*5}, Tadahumi NISHIMURA^{*5}, Daisuke SATO^{*5}

ABSTRACT

The purpose of this survey is to develop a calculation method of energy consumption considering the effect of latent heat load handled by air-conditioning systems. First, we gathered knowledge on lifestyle behavior patterns, building characteristics, latent heat load generation condition for both residential and non-residential buildings and developed a thermal load calculation method that takes latent heat load into consideration properly. Next, we also conducted laboratory experiments on desiccant air-conditioning systems and room air conditioners in dehumidification modes in order to clarify the load characteristics on the moisture absorbing and releasing characteristics for each internal heat load condition and outdoor air condition. Based on these results, we clarified the latent heat load processing mechanism and developed an evaluation method of air conditioning systems considering the effect of latent heat load.

Building Research Institute^{*1}, National Institute for Land and Infrastructure Management^{*2},
Kyoto University^{*3}, Architectural Environment Solutions Inc.^{*4},
DAIKIN INDUSTRIES, LTD.^{*5}

目 次

1	はじめに.....	1
1.1	実施体制	2
1.2	実施内容	3
2	壁体・家具等の吸放湿を考慮した温度計算方法の構築.....	5
2.1	調査の目的	5
2.2	家具・収納物使用実態調査.....	5
2.3	家具・収納物の吸放湿特性の文献調査と測定実験.....	12
2.4	家具・収納物の熱・湿気モデルの作成.....	14
2.5	まとめ	19
3	各種空調システムの潜熱処理メカニズムの解明と機器モデルの構築.....	21
3.1	調査の目的	21
3.2	エアコンの潜熱処理特性把握および潜熱処理モデルの作成.....	94
3.3	ビルマルチエアコンの潜熱処理特性の把握.....	144
3.4	全熱交換機の熱交換能力特性の把握.....	196
3.5	ヒートポンプデシカント調湿外調機特性の把握.....	240
4	住宅における居住者の通風・冷房行為に関する 実態把握と行為モデルの構築.....	275
4.1	調査の目的	275
4.2	WEB アンケート調査に基づく通風・冷房行為のパターンとその決定要因の整理 ..	276
4.3	フィールド実測調査に基づく窓・エアコン操作の生起条件の分析.....	294
4.4	通風・冷房行為に関するモデルの考案.....	311
4.5	まとめ	313
5	潜熱負荷を適切に考慮した熱負荷計算法の開発と基準原案の作成.....	315
5.1	プログラムの概要.....	315
5.2	各クラスの概要	317
5.3	計算結果	322
6	まとめ.....	329

付録 A

東京 A ビルにおけるヒートポンプデシカント調湿外調機の性能特性式から計算される消費電力、全熱能力、潜熱能力と実測値の比較データ（夏季）……………付録 A-1

付録 B

東京 A ビルにおけるヒートポンプデシカント調湿外調機の圧縮機を入力項とする性能特性式から計算される消費電力、全熱能力、潜熱能力と実測値の比較データ……………付録 B-1

付録 C

住宅における居住者の通風・冷房行為に関する実態把握と行為モデルの構築期間実測調査による住宅の夏期室内温熱環境に関する実態データの収集……………付録 C-1

付録 D

蒸暑地実住宅におけるエアコン冷房挙動の実測調査……………付録 D-1

付録 E

家具の吸放湿特性実験……………付録 E-1

付録 F

プログラムソースコード……………付録 F-1

※ 付録 A～付録 F については、本資料(印刷版)には掲載しておりません。

下記の URL にて pdf ファイル版をご覧ください。（平成 31 年 4 月現在）

<http://www.kenken.go.jp/japanese/contents/publications/data/190/index.html>

1 はじめに

省エネルギー基準の暖冷房・空調一次エネルギー計算において算定される潜熱負荷は、建物特性や居住者の行動に起因する要素も、それを実際に処理する空調設備の挙動も、十分に知見が蓄積されているとは言い難く、簡略化して扱われているのが現状である。本調査では、主に冷房エネルギー消費量評価の枠組みにおける潜熱負荷の評価を精緻化するための調査・測定と、それらをふまえて各種潜熱負荷要素のインプット・アウトプット関係の整理を含むモデル化を行うとともに、試算用にプログラムを構築し、これを利用して省エネ基準のモデル建物を対象として冷房一次エネルギー消費量を試算することを目的とする。

1.1 実施体制

本研究の実施体制を図1に示す。協力の4氏には、アンケート・実験・実測等の実施計画策定の補助と助言、およびその実施（学生の協力を含む）を依頼した。

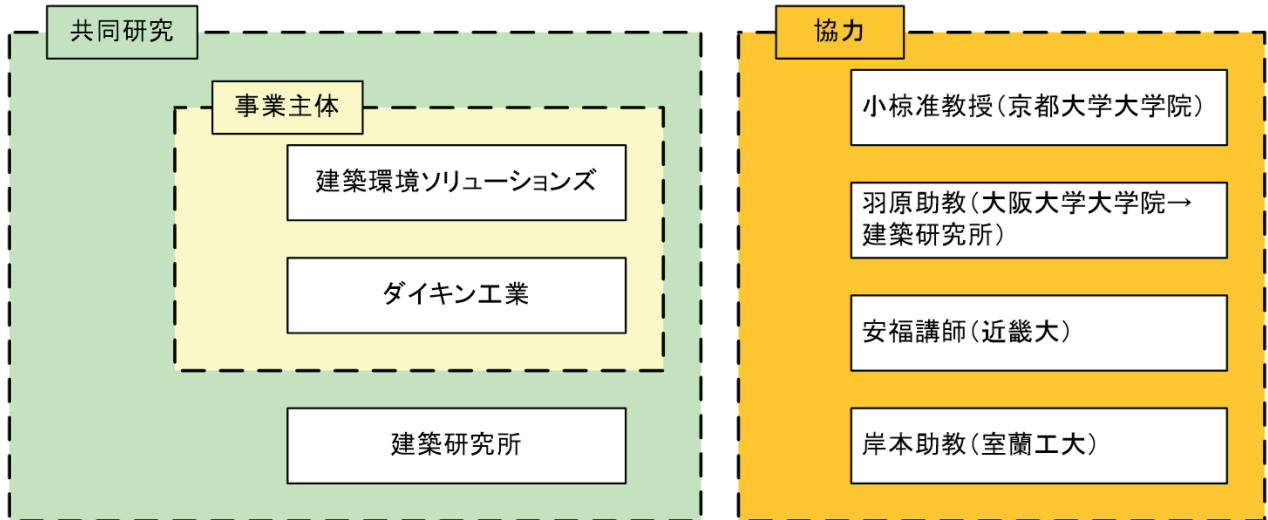


図 1 本研究の実施体制

1.2 実施内容

潜熱負荷やその処理に要するエネルギー消費量に影響を与える要因として、

- ・建物要因
- ・人体要因
- ・設備要因

の3つを考え、それぞれの計算モデル化に必要な要素を実験・実測・調査等により把握することが本調査の主な内容である。

建物要因としては、従来の知見が少ない家具・収納物等の所有量を得るための調査を行った。また、寝具等や吸放湿パネルを用いて実験室実験を行い、室の温湿度変動に与える影響を調査した。

人体の要因としては、主に住宅を対象に居住者の冷房行為実態、すなわちどのような条件で冷房・除湿等をONあるいはOFFするか、あるいは窓を開けるかという点に関して知見を得るためのアンケート調査や被験者実験を行った。

設備要因としては、非住宅の各種設備、住宅用エアコンの顕熱・潜熱処理に関するモデル化を行った。非住宅についてビル用マルチエアコン、デシカント空調、全熱交換器を主な対象とし、実建物において各種空調設備の実働実験を行い、一部補足のために実験室実験も行った。住宅用エアコンについては実験室実験と、潜熱負荷が大きくなることを期待して蒸暑地（沖縄県宮古島）における実測を併せて行った。また、エアコンについてはいったん除去した水分が再度室内側に水蒸気として放出される現象が発生するため、その実験室実験も行った。

これらの調査・実測等で得られた知見を総合して、冷房エネルギー消費量計算プログラムを実装し、プログラム開発内容と住宅の計算結果（従来の冷房一次エネルギー基準値が今回の調査により得られたモデルでどのように変わるか）について記した。

2 壁体・家具等の吸放湿を考慮した温度計算方法の構築

2.1 調査の目的

住宅の温熱環境を快適に維持するために、断熱性・気密性の確保による適切な建築計画に加え、エアコンなどの設備に対して高効率機器を用いることにより、暖冷房消費エネルギーを抑制することが重要といえる。特に冷房において形成される室内温湿度環境を考えた場合、一般的に間欠的使用が多く、室内の収納物の熱・湿気容量の影響が無視できない。これらが、どの程度収納されており、それが室内の温湿度形成や、熱負荷にどの程度影響を与えているのかは明らかになっていない。

本研究では、住宅内の家具・収納物が温度・湿度の挙動、熱負荷にどのような影響を及ぼすのかについて明らかにするため、机、本、衣類、布団、ベッドなどの家具・収納物について以下の様な検討を実施した。

- 1) 家具・収納物の使用実態調査
- 2) 家具・収納物の吸放湿特性の文献調査
- 3) 家具・収納物の熱・湿気モデルの作成

2.2 家具・収納物使用実態調査

2.2.1 WEB アンケートの概要

居住者の行動実態調査で行った WEB アンケートに、家具の使用実態調査を加えた。以下で WEB アンケートの概要を示す。調査対象地域は、国内 3 県から選定する。選定するにあたり高温多湿である太平洋側の気候に着目するとともに、蒸暑地域の中でも地域の差が出るように四国地方、九州地方、沖縄地方を選択した。各地方からの県の選定に関しては標準年を用いてエアコン（冷房、除湿、送風）を利用すると考えられるおおよそ 5 月上旬～9 月下旬頃に特に高温多湿である高知県、鹿児島県、沖縄県の 3 県を対象地域として決定した。アンケート調査は、WEB アンケートで 2013 年 11 月 1 日（金）～8 日（金）の計 8 日間を調査実施日として行った。

アンケートにおいては、対象室をリビングと寝室に対して、住宅の形式（戸建・集合）、構造（RC、木造、鉄骨）、室の広さ、床の仕上げ（フローリング、カーペット、畳）、各部屋の家具の個数（机、椅子、ソファ、棚、座布団、クッション、じゅうたん、布団、ベッド、植物、本）などの情報を得た。また、これに加えエアコンの使用実態と、室内の湿度がもたらす問題として代表的なカビ・結露の実態についてもアンケートを実施した。

2.2.2 紙面アンケートの概要

家具・収納物の使用実態調査を、家具・収納物の設置状況の把握も可能となるように、図面に記述してもらうことを含めて、実施した。調査は、近畿大学の建築学部の学生 544 名に対して、2014 年 1 月中旬に実施し、有効な回答は 503 であった。

質問項目は、以下の表の通りであり、データについて集計を行った。

表 1 家具・収納物のアンケート調査（紙面）

大項目	中項目	小項目
1. 住宅の基本特性		1-1. 都道府県、1-2. 人数構成、1-3. ペットの有無、1-4. 平日、休日の人がいる時間帯・就寝の時間帯、1-5 就寝場所、1-6 主な調理時間帯・時間、1-7. 住宅の建設時期、1-8. リビング・寝室を改修の有無、1-9. 改修理由、1-10. 居住年数、1-11. 何階建ての戸建て住宅か、集合住宅では何階中何階か、1-12. 各部屋の広さ、1-13. 各部屋の階、1-14. 建物の構造、1-15. 建物の断熱、1-16. 断熱箇所、1-17. 窓のガラスの種類・単層/複層、1-18. 各部屋の床暖房の有無、1-19. 各部屋(室内)の床・壁・天井の表面の主な材料
	2. リビングの家具の配置	図面に記入
3. リビングの家具/収納物の種類・大きさ・素材	A 窓・カーテン	A-1 窓の大きさ、A-2 カーテン(レース・遮光)の有無、A-3 遮光カーテンの素材、A-4 カーテンの開閉頻度(夏、冬)
	B. カーペット	B-1 夏におけるカーペットの有無、B-2 カーペットの大きさ、B-3 素材、B-4 冬におけるカーペットの有無、B-5 カーペットの大きさ、B-6 素材
	C ベッド	C-1 大きさ、C-2 種類、C-3 ベッド下収納、C-4 マットレスの有無、C-5 布団の収納方法、C-6 布団(夏)の素材(掛布団)、C-7 布団(夏)の素材(敷布団)、C-8 布団(冬)の素材(掛布団)、C-9 布団(冬)の素材(敷布団)
	D 布団	D-1 大きさ、D-2 布団の収納方法、D-3 布団(夏)の素材(掛布団)、D-4 布団(夏)の素材(敷布団) D-5 布団(冬)の素材(掛布団)、D-6 布団(冬)の素材(敷布団)
	E ソファ	E-1 種類、E-2 大きさ、E-3 素材、E-4 配置
	F 棚(本棚・食器だな等)	F-1 大きさ、F-2 スライド、F-3 奥行、F-4 用途、F-5 扉の有無、F-6 背板、F-7 素材、F-8 配置、F-9 本の場合、収納率
	G タンス(チェスト)	G-1 大きさ、G-2 奥行、G-3 素材、G-4 配置、G-5 収納率(夏)、G-6 収納されている衣類の種類(夏)、G-7 収納率(冬)、G-8 収納されている衣類の種類(冬)
	H 衣類(室内ハンガー掛け)	H-1 衣類カバーなしの衣類の数(夏)、H-2 衣類の種類(夏)、H-3 衣類カバーをかけている衣類の数(夏)、H-4 衣類の種類(夏)、H-5 衣類カバーの種類(夏)、H-6 衣類カバーなしの衣類の数(冬)、H-7 衣類の種類(冬)、H-8 衣類カバーをかけている衣類の数(冬)、H-9 衣類の種類(冬)、H-10 衣類カバーの種類(冬)
	I テーブルについて(リビング)	I-1 テーブルの種類、I-2 テーブルの素材
	J 収納空間について	J-1 収納空間の大きさ、J-2 主な収納物、J-3 収納率、J-4 収納空間の種類・戸、戸の開閉状況
4. 寝室の家具の配置		図面に記入
5. 寝室の家具/収納物の種類・大きさ・素材	3. と同じ	3. と同じ
6. リビングの加湿・除湿源	換気口	6-1 換気口の有無と、有りの場合図面に位置を示す、6-2 換気口が開いている時間帯(夏・冬)
	換気扇	6-3 換気扇の有無と、有りの場合図面に位置を示す、6-4 換気扇が開いている時間帯(夏・冬)
	部屋干し	6-5 リビングで部屋干しを行う季節、6-6 部屋干しを行う理由
	加湿に関する機器類	①開放型暖房器*、②食洗機、③炊飯器、④湯沸しポット、⑤水槽、⑥観葉植物、⑦その他
	エアコンの除湿運転	6-8 エアコンの除湿運転の有無、6-9 使用時期、6-10 使用頻度、6-11 除湿機能を使用する時、6-12 除湿機能の使用理由
除湿器	6-13 除湿器の有無、6-14 使用時期、6-15 使用頻度、6-16 除湿器を使用する時、6-17 除湿器の使用理由	
加湿器	6-18 加湿器の有無、6-19 使用時期、6-20 使用頻度、6-21 加湿器を使用する時、6-22 加湿器の使用理由	
7. 寝室の加湿・除湿源	6. と同じ	6. と同じ
8. 結露・カビの発生	リビング	8-1 結露の発生場所について図面に示す、8-2 結露の発生頻度、8-3 結露の発生時間帯、8-4 カビの発生場所について図面に示す
	寝室	リビングと同じ
	結露の発生状況と対策	8-9 結露の発生を気にするか否か、気にする場合の理由、8-10 結露発生防止の実施の有無、8-11 結露発生防止の対策、8-12 結露水は拭き取り
	カビの発生状況と対策	8-13 カビの発生を気にするか否か、気にする場合の理由、8-14 カビ発生防止の実施の有無、8-15 カビ発生防止の対策、8-16 カビは拭き取り

2.2.3 紙面アンケートの結果の分析

図2にアンケート対象者の住まいの府県、住宅形式と家族構成を示す。大阪、兵庫、奈良、京都の順に、近畿エリアの住まいだけで7割強を占めている。アンケートを実施した大学が大阪府下であることが影響していると考えられる。住宅の形式は、73%が戸建住宅、25%が集合住宅であった。

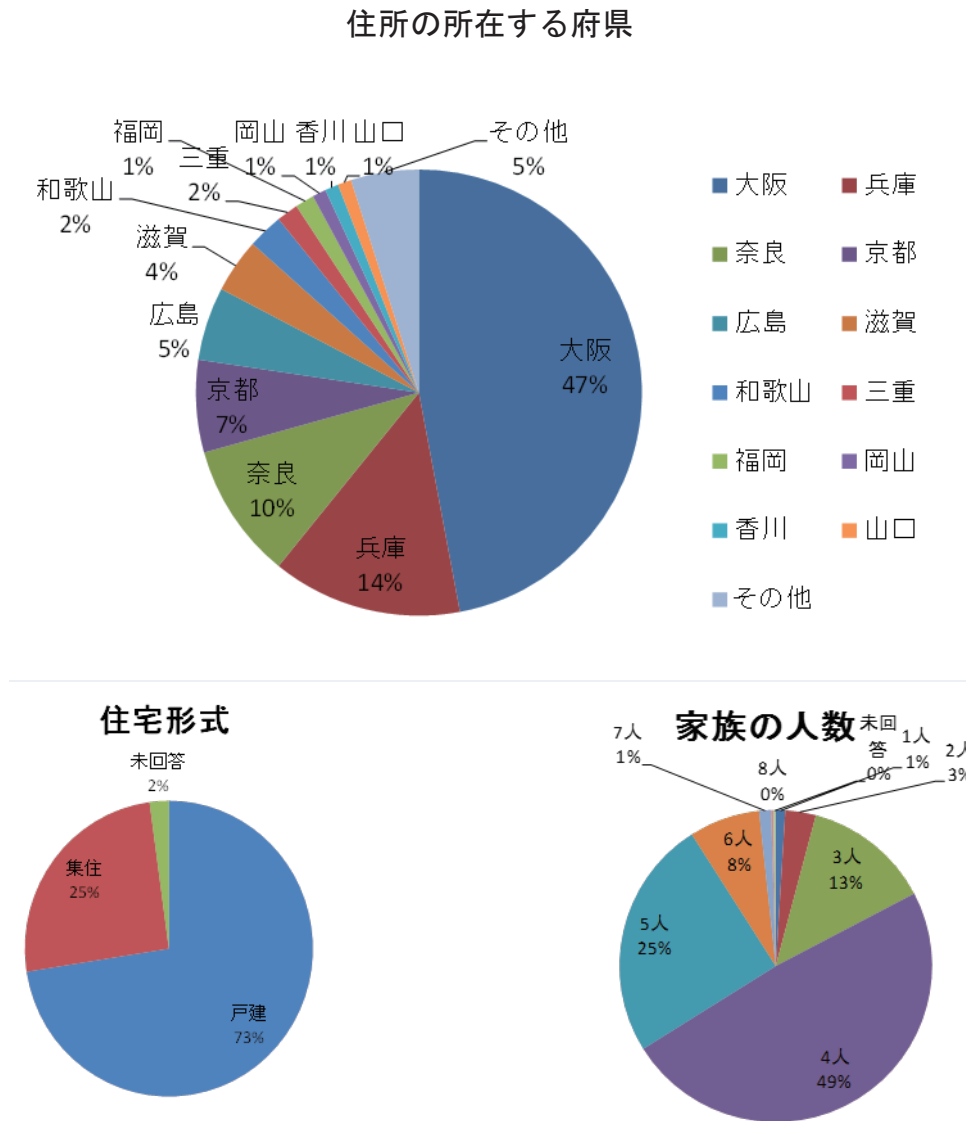


図2 住所の所在する府県、住宅形式と家族構成

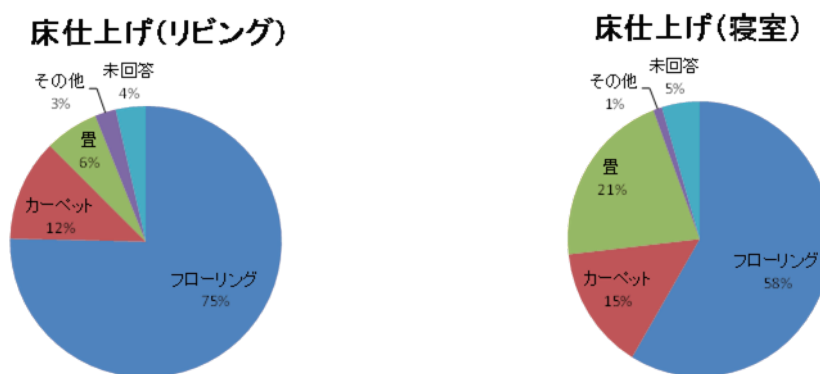
2.2.3.1 仕上げ

(1) 床仕上げ

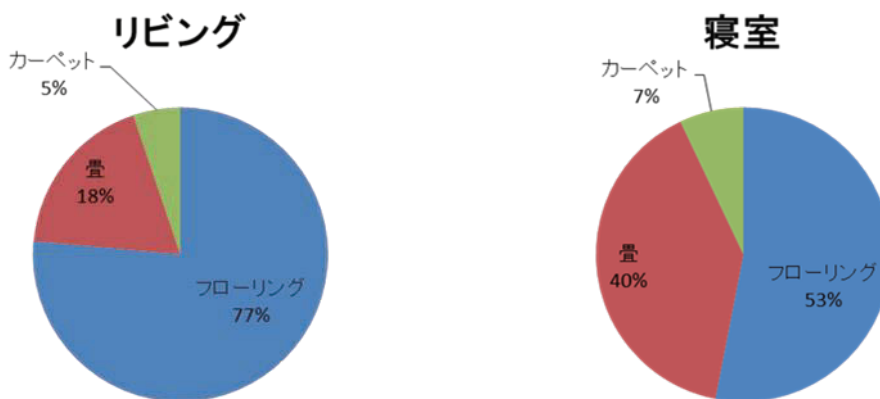
床表面の主な仕上げについては図3に示す通りである。

リビングの床表面仕上げが「フローリング」と答えた方は75%、「畳」と答えた方の割合は6%、寝室ではそれぞれ58%、21%である。いずれの部屋もフローリングが多く、リビングより寝室の方が畳の割合が高い。「カーペット」と答えた方の割合はリビングでは12%、寝室では14%と約1割を超える程度である。

平成26年度に実施した高知、鹿児島、沖縄の3県のWEBアンケート結果と比較するとフローリングの割合はほぼ等しいが、畳の割合は、高知、鹿児島、沖縄の方が割合が多かった。



(a) 平成27年度調査結果



(b) 平成26年度調査：高知、鹿児島、沖縄

図3 床仕上げの分布

(2) 壁仕上げ

壁面の内装仕上げについてはリビングと寝室で大きな差はなくビニル、布、紙クロスが7割程度を占めている(図 4)。

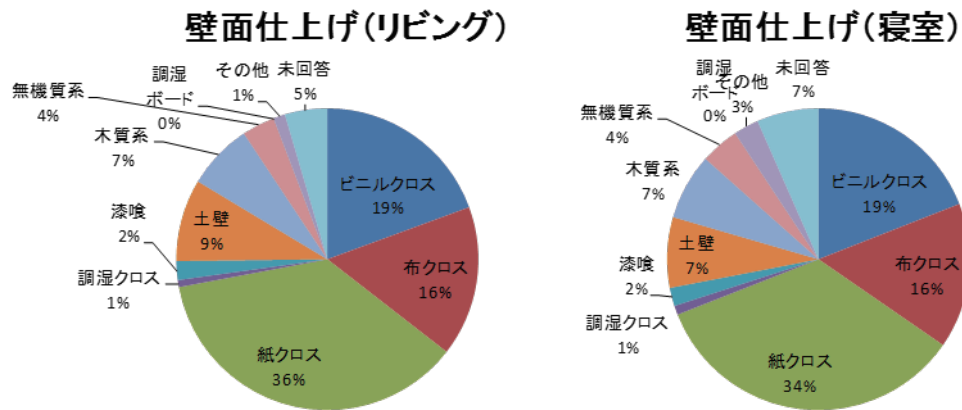


図 4 壁仕上げの分布

2.2.3.2 広さ

リビングの広さは8~13畳程度が多く、平均は12.6畳、寝室は6畳程度が多く、平均は7.5畳であった(図 5)。

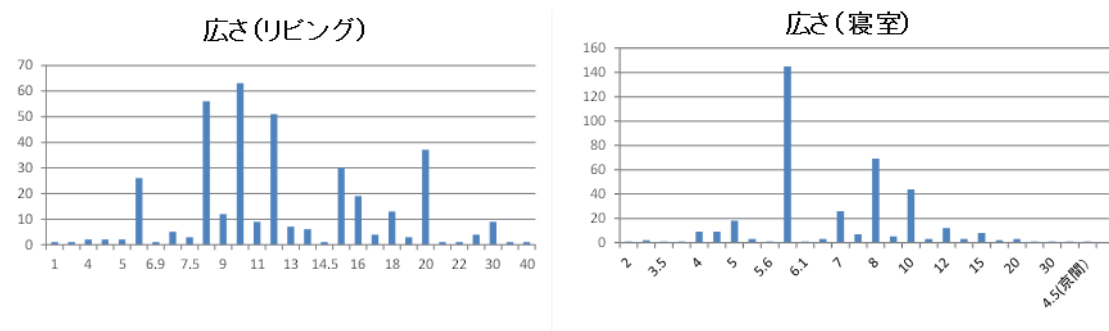


図 5 部屋の広さ

2.2.4 集計の結果

2.2.4.1 棚

リビングの棚の平均数は2.1であった。用途としては食器棚が最も多くサイズは大きなもの(C)が多かった(図 6、図 7、図 8、図 9)。

寝室の棚の平均数は1.0であった。用途としては本棚が最も多く8割以上を占める。サイズは(D)が多かった(図 10、図 11、図 12)。

棚の数(リビング)

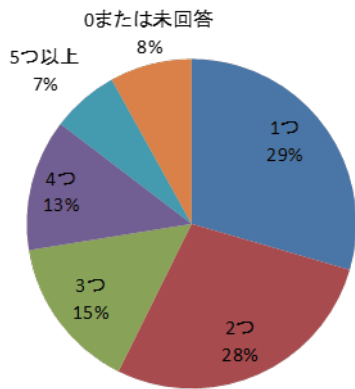


図 6 棚の個数

用途(リビング)

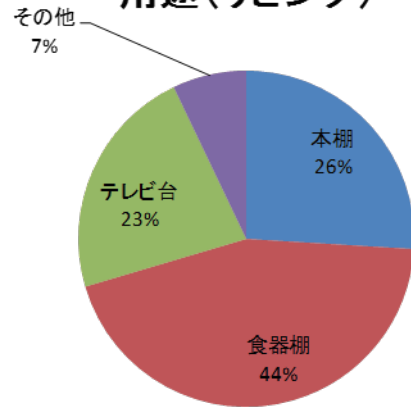


図 7 棚の用途

棚のサイズ(リビング)

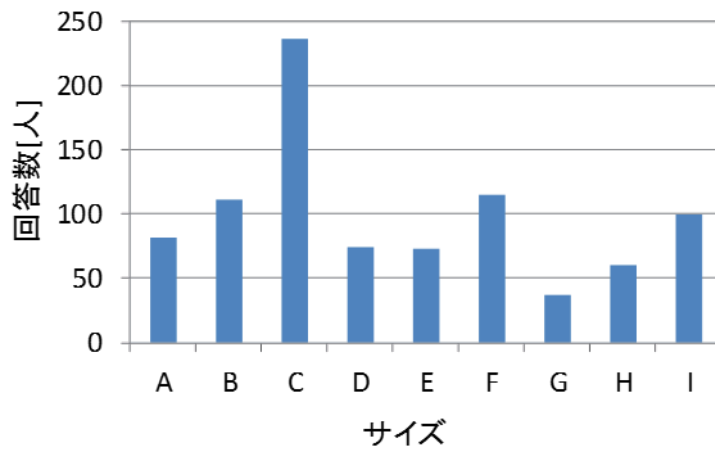


図 8 棚の大きさ

高さ / 幅	50cm 程度	90cm 程度	120cm 程度
180cm 程度	A	B	C
100cm 程度	D	E	F
50cm 程度	G	H	I

図 9 棚の大きさの対応関係

棚の数(寝室)

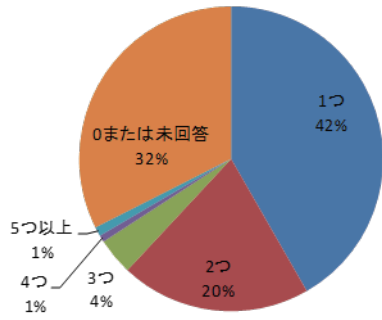


図 10 棚の個数

用途(寝室)

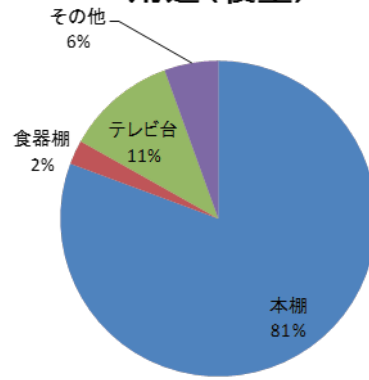


図 11 棚の用途

棚のサイズ(寝室)

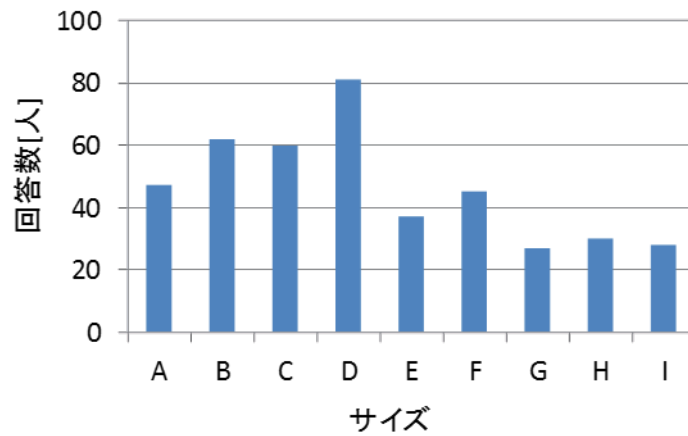


図 12 棚の大きさ

2.3 家具・収納物の吸放湿特性の文献調査と測定実験

家具・収納物における吸放湿特性に関わる物性値の文献調査を実施した。表 2 に、熱湿気物性値として密度 ρ [kg/m³]、熱伝導率 λ [W/mK]、湿気伝導率 [g/ms(g/kg')], 容積比熱 C_p [J/m³K] を、図 13 に平衡含水率を示す。

表 2 家具・収納物の熱・湿気物性値^{i, ii, iii, iv, v, vi, vii, viii, ix}

材料	ρ [kg/m ³]	λ [W/mK]	λ' [g/ms(g/kg')]	C_p [J/m ³ K]
布団(羽毛)	7.7	0.056	1.39E-05	1.37E+04
布団(羊毛50% ポリエステル50%)	11.8	0.056	1.97E-05	1.58E+04
布団(綿60% ポリエステル40%)	18.5	0.056	1.67E-05	2.48E+04
衣類(綿)	81	0.056	2.78E-05	1.08E+05
カーテン(レーヨン)	316	0.079	1.39E-05	4.29E+05
紙(断面)	686	0.209	3.89E-06	8.92E+05
紙(表面)	686	0.209	5.00E-07	8.92E+05
カーペット(羊毛100%)	273	0.080	1.74E-02	3.18E+05

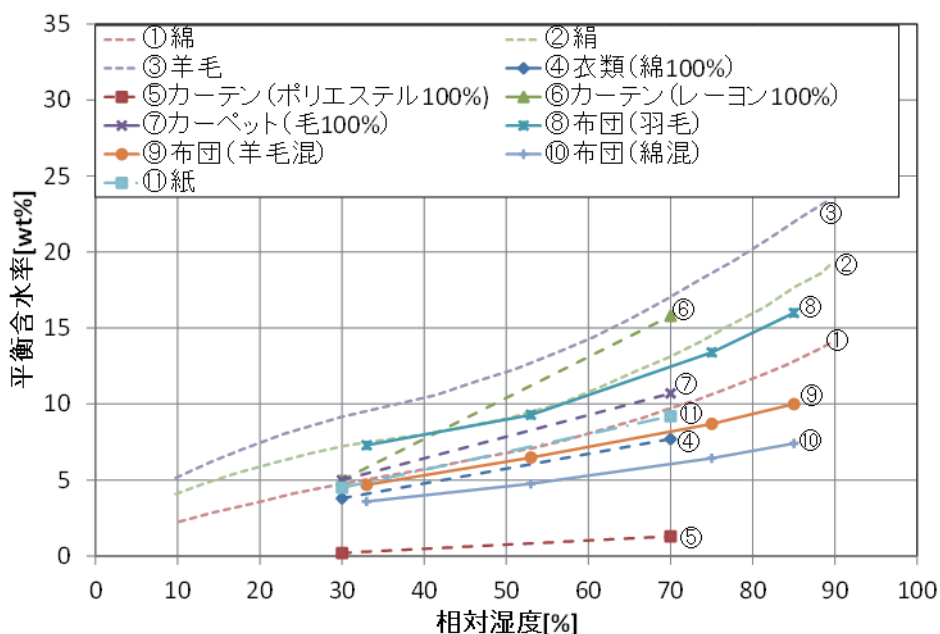


図 13 家具・収納物の平衡含水率^{I-IX}

上記、平衡含水率は相対湿度 30～80%程度ではほぼ直線と見なせるため、近似直線式を下式で示す様に作成した。

$$\varphi = \frac{\rho_d}{\rho_w} (a + bh)$$

式 1

ただし、

φ	: 体積基準平衡含水率	[m ³ /m ³]
ρ_d	: 乾燥密度	[kg/m ³]
ρ_w	: 水密度	[kg/m ³]
h	: 相対湿度	[-]

平衡含水率の相対湿度勾配は下式となる。

$$\frac{d\varphi}{dh} = \frac{\rho_d}{\rho_w} b$$

式 2

得られた係数**b**を表 3 に示す。

表 3 各種材料の平衡含水率の相対湿度勾配の係数

材料	b
衣類 (綿)	0.0975
カーテン (レーヨン 100%)	0.27
カーペット (毛 100%)	0.1425
布団 (羽毛)	0.1669
布団 (羊毛混)	0.1101
布団 (綿混)	0.0731
紙	0.1175

2.4 家具・収納物の熱・湿気モデルの作成

建築壁体内における熱と水分の移動に関する理論は、松本^x、池田^{xi}により確立しており、家具も同様の取扱が可能と考えられる。1次元の壁体の蒸気拡散支配の熱水分同時移動方程式は下式の通りである。

$$(c\rho + Lv) \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) + L\kappa \frac{\partial X}{\partial t} \tag{式 3}$$

$$(c'\rho' + \kappa) \frac{\partial X}{\partial t} = \frac{\partial}{\partial x} \left(\lambda' \frac{\partial X}{\partial x} \right) + v \frac{\partial T}{\partial t} \tag{式 4}$$

ここで、

c	: 材料の比熱	[J/kgK]
ρ	: 材料の密度	[kg/m ³]
L	: 水の相変化熱	[J/kg]
v	: 含水率の絶対湿度勾配による吸放湿特性値 (ニューウ)	[kg/ m ³ K]
T	: 温度	[°C]
t	: 時間	[s]
λ	: 熱伝導率	[W/mK]
κ	: 含水率の温度勾配による吸放湿特性値 (カッパ)	[kg/ m ³ (kg/kg')]
ρ'	: 乾燥空気密度	[kg'/ m ³]
λ'	: 湿気伝導率	[kg/ ms(kg/kg')]

家具・収納物の吸放湿については、図 14 に示すような1次元壁体へのモデル化を行った。容積 V [m³]を空気に接する部位の表面積 L [m²]で除して、材料の厚み L [m]を算出する。

$$L = \frac{V}{S} \tag{式 5}$$

また、家具・収納物の室内側表面からの厚み L [m]の位置では断熱・断湿とする。

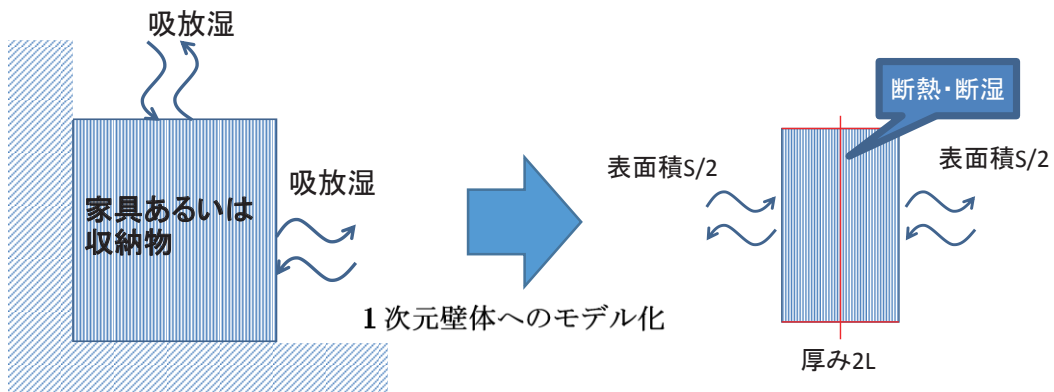


図 14 家具あるいは収納物の1次元壁体へのモデル化

上記の様な家具モデルの実測との比較については、梅野らⁱⁱⁱが、放湿過程についてやや放湿量を過大に見積もるものの1次元解析モデルの妥当性を確認している。家具・収納物を、空気側の伝達に寄与する部位に対して、1次元の熱水分移動現象として取り扱ってよいと考えられる。

以下では、上記モデルを用いた家具がある部屋の吸放湿の影響の数値シミュレーションによる検討例を示す。図15に示すように解析対象は集合住宅の一室を想定し、本棚の本、ベッド（綿）を吸放湿材とする。なお室内側の表面の吸放湿性は無視する。比較のため家具なしの条件も行う。計算条件は、表4に示す通りである。アンケート調査から求めた寝室の平均的な本棚サイズ、ベッドサイズを想定している。

表4 計算条件

換気回数 [回/h]	家具	外気[°C]	外気絶対湿度 [g/kg]	室温	室相対湿度	室内水分発生量 [g/h]	室内熱発生量[W]
0.5	本棚の本 (1.8×1.2×0.2) ベッドの綿 (1×2×0.2)	29±3	16.32一定	27°C (17-22時の間)	60% (17-22時の間)	40 (17-22時の間)	60 (17-22時の間)

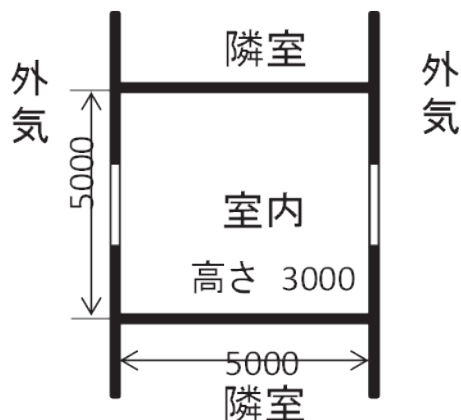


図15 解析対象(単位: mm)

温湿度の計算結果を図 16 図 17 図 18 に示す。図中の吸放湿ありが家具ありの条件を、吸放湿なしが家具なしの条件を表す。図より家具のみの有無でも室温湿度が多少異なる変動を示す。家具ありの場合、エアコン停止後の湿度上昇が最大で 3%程度抑制されている。

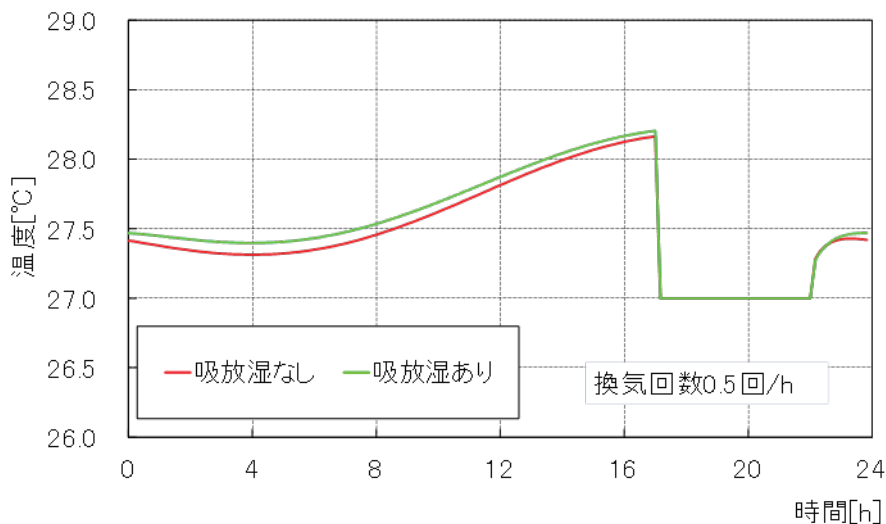


図 16 家具の有無による温度の比較

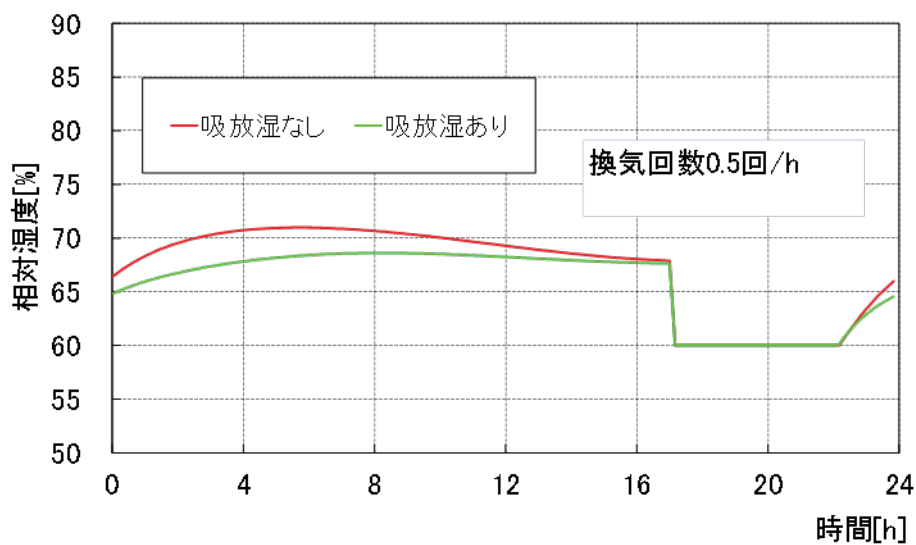


図 17 家具の有無による相対湿度の比較

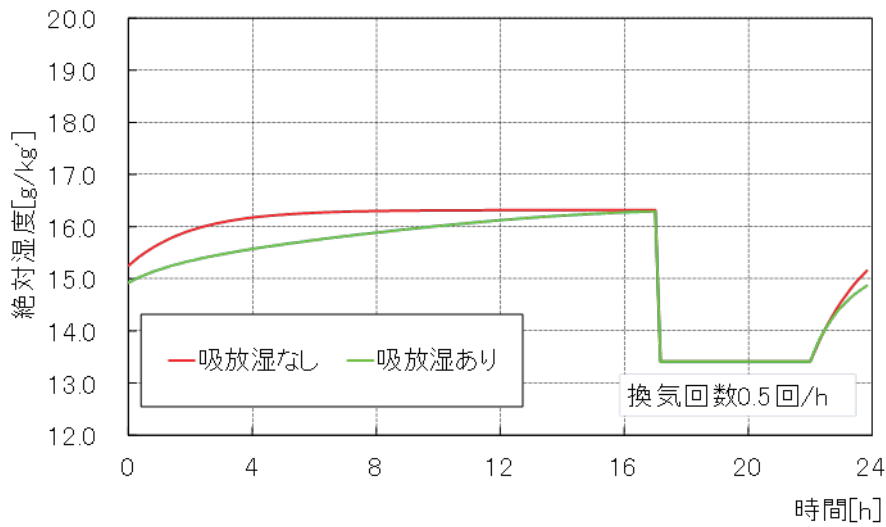


図 18 家具の有無による絶対湿度の比較

次に家具の有無による全熱負荷の比較を、図 19 に、家具がある場合とない場合の顕・潜熱負荷をそれぞれ図 20、図 21 に示す。図より夕方に冷房を行う場合、室内の吸放湿性を考慮することで、全熱負荷が、4%程度増加することが分かる。以上から、平均的な家具設置条件で熱負荷は増えるが、家具の吸放湿が熱負荷へ与える影響は大きくはなかったといえる。

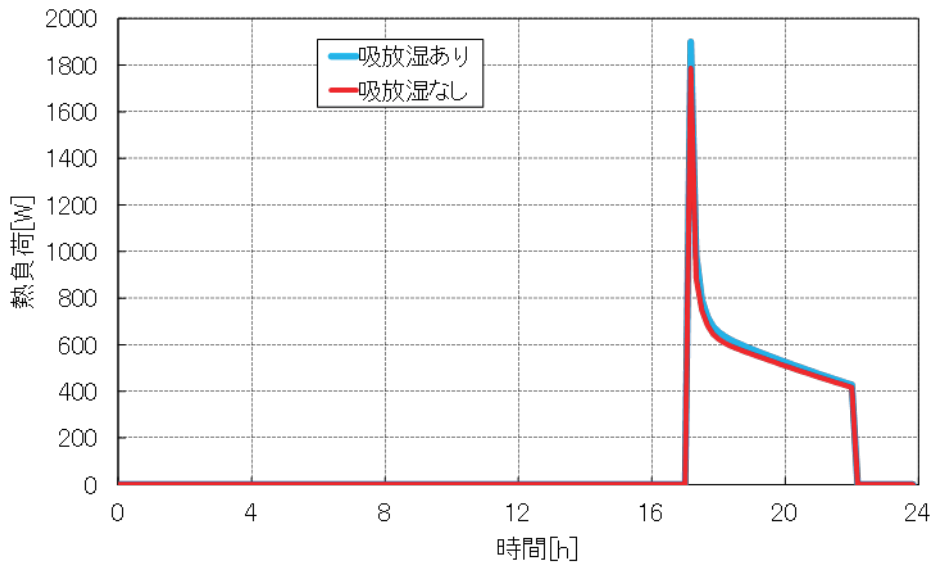


図 19 家具の有無による全熱負荷の比較

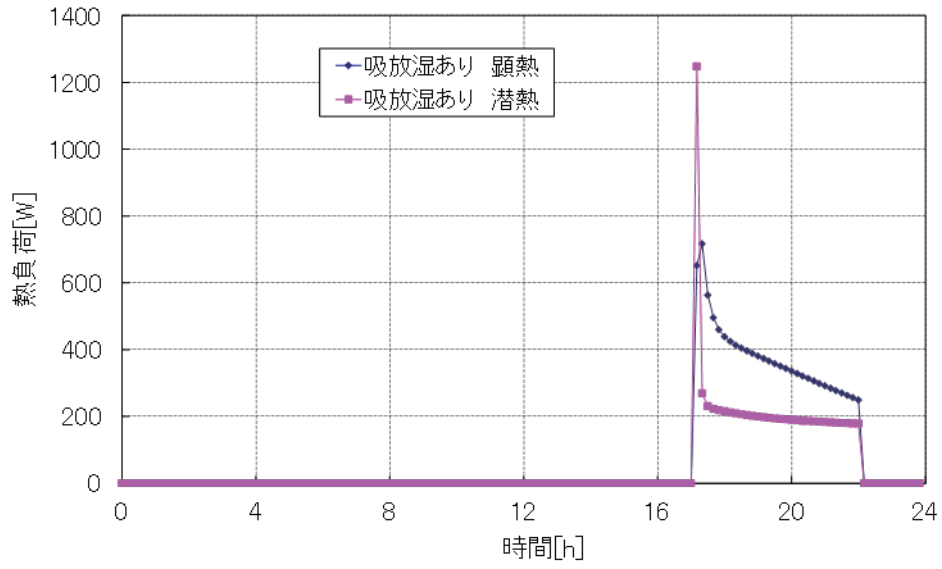


図 20 家具ありの場合の顕熱、潜熱負荷

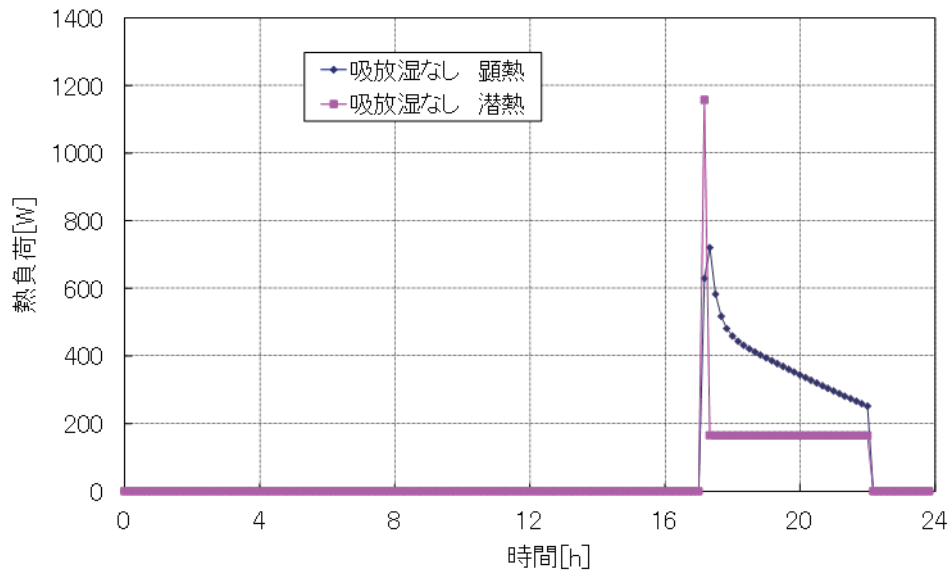


図 21 家具なしの場合の顕熱、潜熱負荷

2.5 まとめ

本研究では、住宅内の家具・収納物が温度・湿度の挙動、熱負荷にどのような影響を及ぼすのかについて明らかにするため、机、本、衣類、布団、ベッドなどの家具・収納物について以下の様な検討を実施した。

- 1) 家具・収納物の使用実態調査
- 2) 家具・収納物の吸放湿特性の文献調査
- 3) 家具・収納物の熱・湿気モデルの作成

得られた結果は以下の通りである。

大阪、兵庫、奈良、京都の順に近畿エリアの住まいで7割強を占める家具・収納物の使用実態に関するアンケート調査結果から、以下の結果が得られた。

- 1) 住宅の形式：73%が戸建住宅、25%が集合住宅であった。
- 2) リビングの平均広さは12.6畳、寝室の平均広さは7.5畳であった。
- 3) 床仕上げは、リビングでは「フローリング」78%、「カーペット」11%、「畳」6%であり、寝室では「フローリング」66%、「畳」19%、「カーペット」15%であった。
- 4) 棚については、リビングでは平均2.1台であり、食器棚、本棚が多く、収納率80%以上が6割あった。寝室では平均1.0台であり、本棚が多く、収納率80%以上が7割あった。

家具・収納物における吸放湿特性に関わる物性値の文献調査を実施し、布団、紙、衣類、カーテン、カーペットの物性値を得た。

家具・収納物の吸放湿を明らかにするため、既存の熱水分同時移動方程式を用いて1次元の壁体として扱うモデル化を行った。また、このモデルを用いて、例題計算として寝室の平均的なベッド、本棚の収納環境、夏期の夕方での冷房使用を想定し、数値シミュレーションを行った。その結果、以下の結果が得られた。

- 1) 家具の有無で室温湿度が多少異なる変動を示す。特に家具ありの場合、エアコン停止後の湿度上昇が最大で3%程度抑制された。
- 2) 室内の吸放湿性を考慮することで、全熱負荷が、4%程度増加した。平均的な家具設置条件で熱負荷が増えるが、その影響は、大きくはなかった。

室ごとの家具・収納物の設置量とこれらのモデルを考慮した室の解析手法を用いることで、エアコンの挙動を考慮した温湿度解析、熱負荷解析、消費電力の解析は可能となる。

3 各種空調システムの潜熱処理メカニズムの解明と機器モデルの構築

3.1 調査の目的

潜熱処理機能を有する空調設備システムの性能評価方法を整備する上で、空調設備システムの実運転状態における性能を予測する手段が必要となる。一方、潜熱処理を有する空調設備システムには冷却除湿、吸着除湿などの種々の原理が存在する。

そこで、潜熱処理を有する空調設備システムの性能予測計算方法をその動作原理毎に整備することを目的として、市場に存在する空調設備システムの類型化を行った。本報告では黄色部分の評価を主に行った。

表 5 分類表

種類	原理	方式		機器	能力制御方法		規格	汎用性	性能データ (潜熱能力)	
		大分類	小分類		大分類	小分類				
除湿	冷却 1-2-1	蒸気圧縮式	-	空調機	蒸発温度	ファン風量固定 ファン風量自動 ファン風量自動	JISC9612(ルームエアコンディショナ) JISB8616(パッケージエアコンディショナ) JISB8615-1(エアコンディショナ第一号: 直吹き形エアコンディショナ及びヒートポンプ 一定格性能及び運転性能試験法)	◎	△	
				外調機	吹出温度制御 吸込温度制御	- -	- -	- -	- -	- -
				除濕機	蒸発温度	-	JISC9617(除濕機)	-	-	-
				再熱	再熱ON/OFF	蒸発温度	JISB8617(除濕機)	-	-	-
				外調機	吹出温度制御	-	-	-	-	-
				外調機	吸込温度制御	-	-	-	-	-
	吸着 1-2-2	ローター式	-	外調機	蒸発温度	-	JISB8628	△~○	○	
				外調機	吸込温度制御	-	-	-	-	
				外調機	ヒーター温度、風量	-	-	-	-	
		パッチ式	-	外調機	蒸気圧縮	蒸発温度	-	-	-	
				外調機	ハイブリッド式	蒸気圧縮:蒸発温度、吸着:ヒーター温度	-	-	-	
				外調機	直接冷却吸着	冷媒循環量	-	-	-	
		膜透過式 1-2-3	蒸気圧差(室内-室外間)	+圧縮(真空)	外調機	全熱交	-	JISB8628	○	○
					外調機	圧縮(1-2-4)	-	-	-	-
					外調機	プロセス用	-	-	-	-
吸収 1-2-5	スプレー式	膜透過	外調機	液温度	-	-	×	×		
			外調機	液温度	-	-	-	-		
加湿	気化式 1-2-6	液下浸透 気化式	加湿器	水ON/OFF	ファン風量	JEMA1426(加湿器)	○	○		
			空調機	水ON/OFF	凝縮温度	-	-	-		
			外調機	水ON/OFF	凝縮温度	-	-	-		
		透過膜式	-	加湿器	水ON/OFF	ファン風量	JEMA1426(加湿器)	○	○	
				空調機	水ON/OFF	凝縮温度	-	-	-	
				外調機	水ON/OFF	凝縮温度	-	-	-	
	水噴霧 1-2-7	超音波	-	加湿器	振動子ON/OFF	-	JEMA1426(加湿器)	△	△	
				空調機	ポンプON/OFF	-	-	-	-	
				外調機	ポンプON/OFF	-	-	-	-	
		電力利用型	-	加湿器	ヒーターON/OFF	-	JEMA1426(加湿器)	△	○	
				加湿器	水ON/OFF	-	JEMA1426(加湿器)	△	○	
				加湿器	水ON/OFF	-	-	-	-	
		全熱交	-	加湿器	全熱交	-	-	-	-	
				加湿器	全熱交	-	-	-	-	
				加湿器	全熱交	-	-	-	-	
膜透過式 1-2-9	吸収	-	加湿器	液温度	-	-	○	○		
			加湿器	液温度	-	-	-	-		
			加湿器	液温度	-	-	-	-		
乾式 脱着 1-2-10	ローター式	-	加湿器	室内空気吸着	ヒーター温度	-	-			
			加湿器	室外空気吸着	ヒーター温度	-	-			
			加湿器	室外空気吸着	ヒーター温度	-	-			
パッチ式	-	加湿器	直接加熱脱着	冷媒循環量	-	-	△	○		
		加湿器	直接加熱脱着	冷媒循環量	-	-	-	-		
		加湿器	直接加熱脱着	冷媒循環量	-	-	-	-		

表に示すように、除湿では、

- ・ 蒸気圧縮式
- ・ 吸着式
- ・ 膜透過式 (全熱交換器)

の採用が多く、加湿では、

- ・ 気化式
- ・ 膜透過式 (全熱交換器)

の採用が多い。

また、蒸気圧縮式では冷房システムとして利用された際も成行きで湿度処理を行う。温度制御方式としては、冷媒温度を可変にする方式と固定にする方式の大きく二つに分類される。また、蒸気圧縮式空調機において住宅用の製品は除湿機能を併設したものが多く存在するためその評価も併せて実施する。

業務用空調においては、換気装置として全熱交換器、更に近年潜熱・顕熱分離空調を行うことを目的とした乾式除湿器の採用が進んでいる。また、加湿器としてはエネルギー消費効率の面から乾式除湿器の採用が主流である。以下に、原理、規格について報告する。

3.1.1 冷房除湿を行う空調設備システム

3.1.1.1 冷却除湿

(1) 動作原理

冷却除湿運転の原理を図 22 に示す。

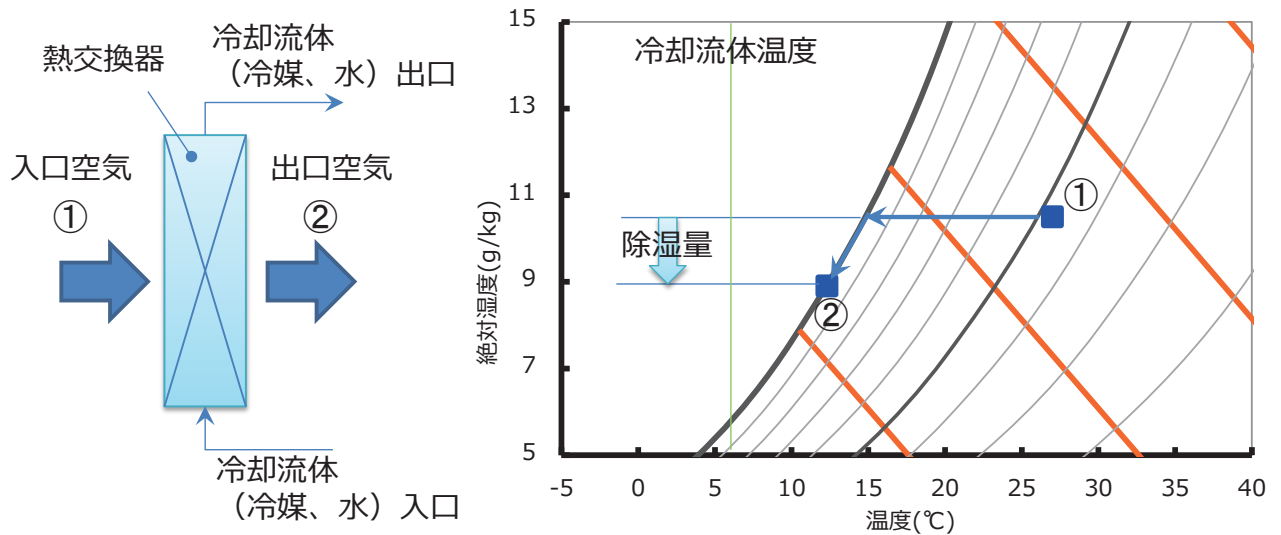


図 22 冷却除湿原理

空気が、熱交換器を流通時に冷却流体と熱交換することによって冷却され、(①→②) 露点温度以下まで冷却されることによって除湿を行う。

出口空気の温度は

- ・ 冷却流体が流通する伝熱管と冷却流体間の熱伝達
- ・ 伝熱フィンフィン効率
- ・ 空気と伝熱フィン間の熱伝達率
- ・ 空気中の熱伝導

の影響で冷却流体温度よりも高い温度となる。クロスフィン型熱交換器の構造を図 23 に、模式図を図 24 に示す。

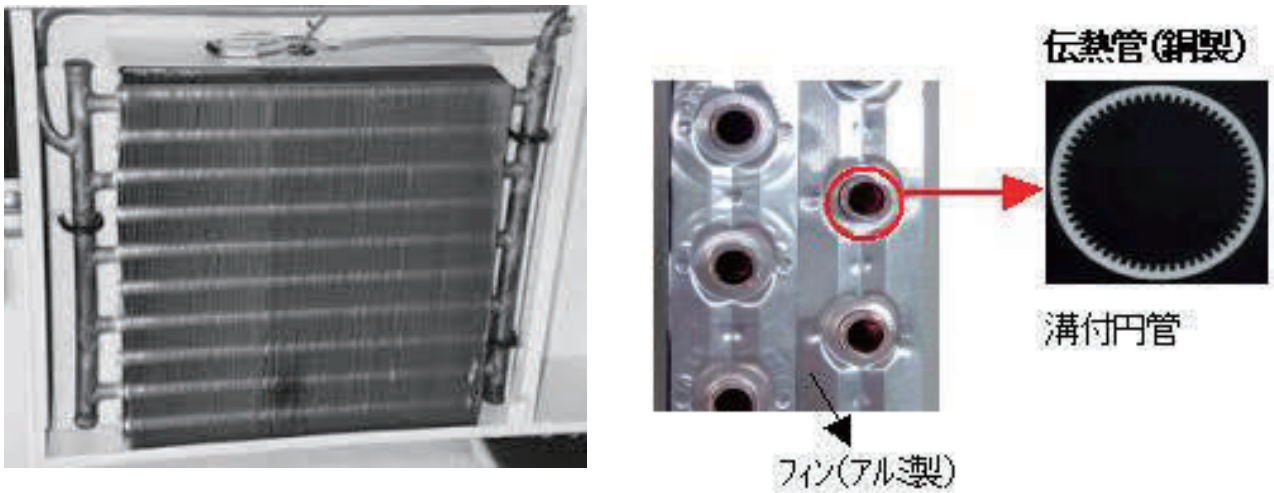


図 23 クロスフィン型熱交換器構造

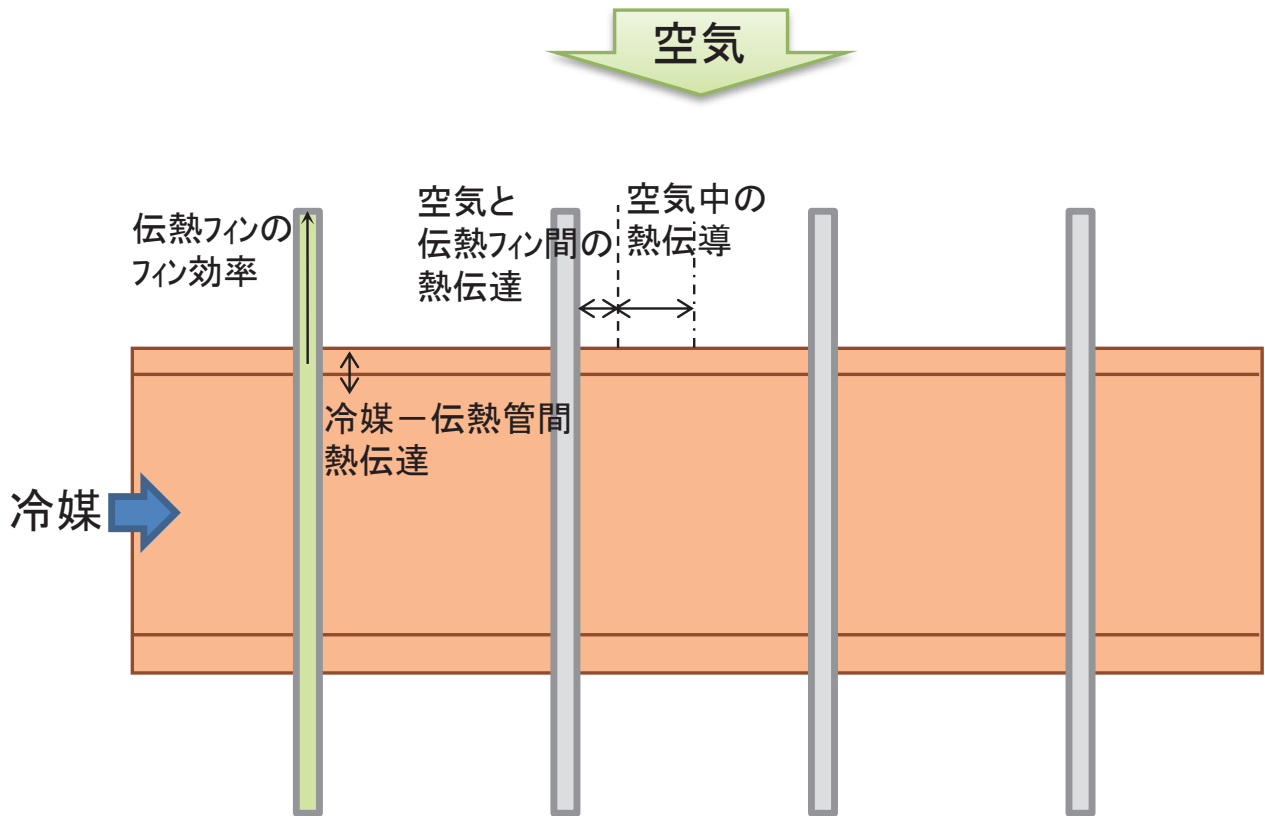


図 24 熱交換器模式図

この時、空気中の熱伝導速度と比較して空気中への水分の物質拡散速度の方が早いため一般的には出口空気は飽和蒸気圧よりも若干低い水蒸気分圧となる。

(2) 冷却除湿を行う空調設備システム

冷却除湿は、一般的な空調設備システムの冷房運転と同時に発生する。一般的な空調設備システムの模式図を図 25 に示す。

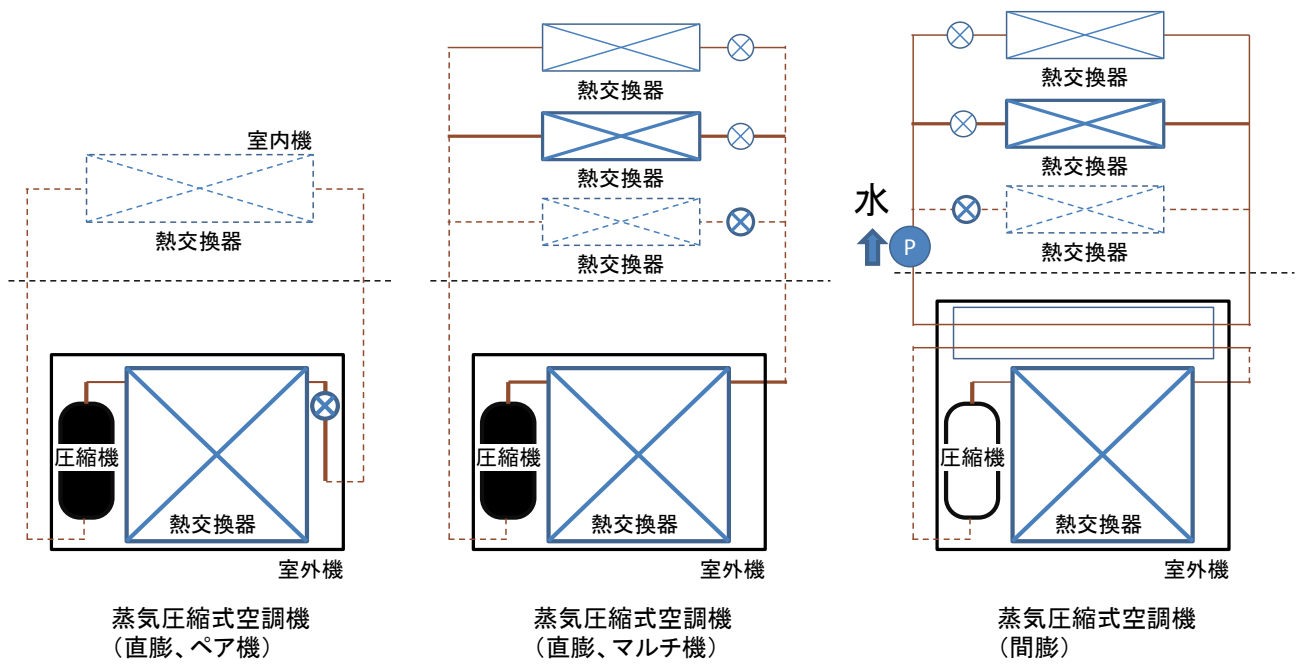


図 25 一般的な空調設備システム

空調機の冷房能力（顕熱能力）制御の方法として、

- ・ 冷却流体の温度（冷媒の蒸発温度）を変化させる方式
- ・ 冷却流体の温度（冷媒の蒸発温度）を固定にして冷媒循環量（冷媒出口過熱温度：Super Heat(SH)）やファン風量を変化させる方式

の二種類が存在する。

各々の運転状況を図 26 の空気線図に示す。

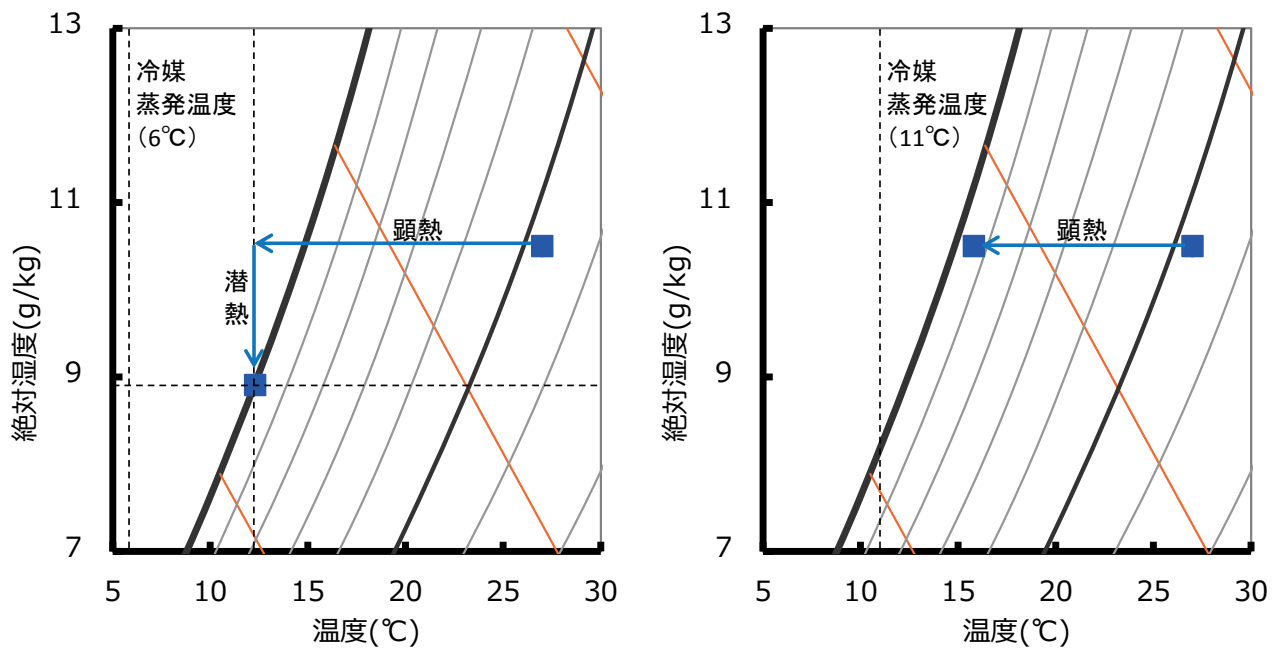


図 26 空調能力制御変化時の空気線図

左図に、冷媒蒸発温度を固定した際の空気線図上の動きを示す。図に示すように、顕熱処理時に同時に潜熱も処理するため除湿量が確保される。従って、室内温湿度が一定の場合には顕熱能力比（顕熱能力/全熱能力）はほぼ一定となる。

右図に、熱能力が小さい際に冷媒蒸発温度を高くして運転を行った際の空気線図を示す。図に示すように、蒸発温度を変化させることによって顕熱能力比（SHF）が変化し、蒸発温度を高くした際（顕熱能力が小さい場合）には潜熱能力が著しく低減する。

一般的には、居住向け建築物に冷房能力を蒸発温度で可変制御する方式、非居住向け建築物に蒸発温度を固定にして冷房能力をファン風量もしくは冷媒循環量で制御をする方式が採用されている。一方、近年エネルギー消費量削減を目的として非居住向け建築物においても蒸発温度を可変制御する方式を採用し、潜熱は別設置の空調設備システムで処理を行う潜熱・顕熱分離空調システムが採用されはじめています。

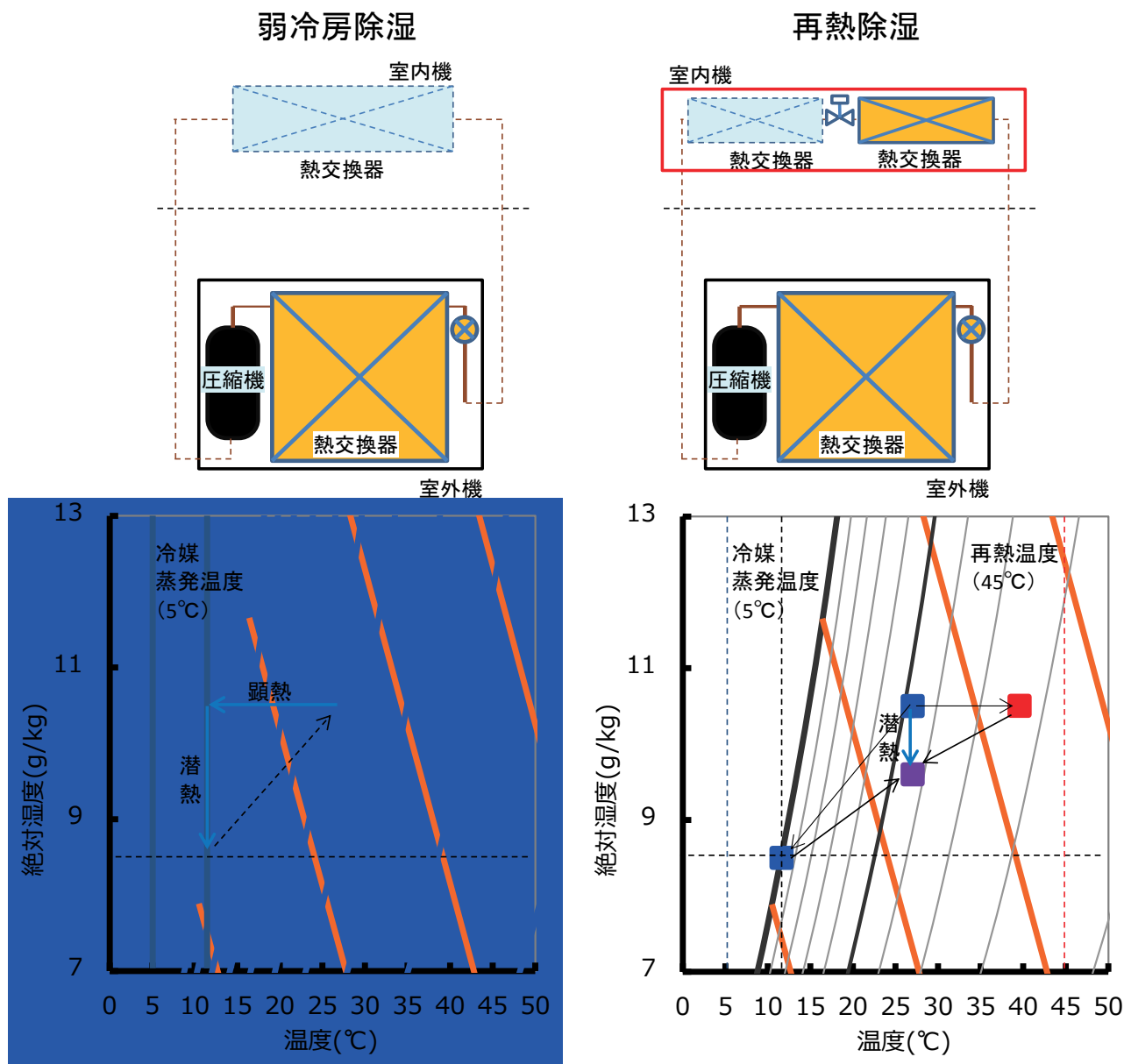


図 27 除湿運転時の空気調和設備模式図および空気線図

冷却除湿の原理を用いた、除湿運転モードを有する空調設備システムが存在する。除湿運転モードを有する代表的な空調設備システムの模式図および空気線図を図 27 に示す。

左図に一般的に弱冷房除湿と呼ばれる除湿運転方式、右図に一般的に再熱除湿と呼ばれる除湿運転方式を示す。弱冷房除湿は、通常の冷房運転時と比較して冷媒の蒸発温度を低く設定し、熱交換器を流通する空気の風量を低減して運転する方式である。空気線図に示すように通常の冷房運転時と比較して小さい顕熱比で運転することによって除湿量を確保する。一方、潜熱を処理する際に併せて顕熱も処理してしまうため除湿運転のみを実施することは不可能である。一般的に弱冷房除湿運転を実施するにはドラフトを低減するために、ファン風量を低減するか、熱交換器の過熱度を大きく取ることによって能力が出ない状態にし、ガス冷媒が流通している熱交換器の部分バイパスした空気を混入することによって吹き出し空気の温度を室温に近づけて運転を実施する。

一方、再熱除湿運転は図 27 の右図に示すように室内機の熱交換器の一部を凝縮器として運転することによって、加熱した空気と冷却した空気を混入した後に室内に吹出し、室内空気を冷却することなく除湿運転を実現する。但し、再熱除湿運転は一度冷却した空気を再度暖めて室内に吹出す運転を実施するために効率が通常の冷房運転と比較して低くなるという課題点が存在する。

(3) 冷却除湿を行う空調設備システムの規格で定められた性能評価点、カタログ、技術資料性能評価点

冷却除湿を行う空調設備システムの性能に関連する規格、メーカー各社の技術資料記載データ、カタログ記載データを表 6 に示す。

表 6 冷却除湿を行う空調設備システムの規格一覧

番号	名称	適用範囲
JISC9612	ルーム エアコンディショナー	定格能力 10kW 以下のルームエアコン ・圧縮式冷凍機,送風機などを一つのキャビネットに内蔵した一体形 ・一台の室外機に一台の室内機を接続した分離形 定格能力 28kW 以下のルームエアコン ・一台の室外機に二台以上の室内機を接続した分離形 マルチ形ルームエアコンディショナー
JISB8616	パッケージ エアコンディショナー	業務用の建物に用いられるよう設計・製作された エアコンディショナーであって、空冷式のものと及び 水冷式冷房専用のもののうち、定格冷房能力が 28kW 以下のもの
JISB8615-1	エアコンディショナー 第 1 部： 直吹き形 エアコンディショナー及び ヒートポンプ ー 定格性能 及び運転性能試験法	・直吹き形空冷式エアコンディショナー、及び 直吹き形空気対空気ヒートポンプ ・定格能力 8kW 未満かつ機外静圧が 25Pa 未満で運転するように 設計した、ダクト形エアコンディショナー 及びダクト形ヒートポンプ ・全ての室内機を常に同時運転するマルチエアコン
JISC9617	電気除湿機	室内の湿度を低下させることを目的とし、圧縮式冷凍機,送風機 などを一つのキャビネットに内蔵したもので、除湿能力を 発揮するのに必要な電力の合計が 500W 以下の電気除湿機 (熱交換器で冷却,除湿した空気を再加熱する方式のもの)

(I) JIS C 9612

表 7 JIS C 9612 試験条件

		室内側吸込空気温度		室外側吸込空気温度	
		乾球温度	湿球温度	乾球温度	湿球温度
定格 条件	標準気候	27℃	19℃	35℃	24℃
	低温気候	21℃	15℃	27℃	19℃
	高温気候	29℃	19℃	46℃	24℃
低温条件		27℃	19℃	29℃	19℃

・表示項目

定格冷房能力[kW]：全熱能力（製造業者が指定する定格冷房能力の97%以上）
定格冷房消費電力[kW]（製造業者が指定する定格冷房消費電力の103%以下）
通年エネルギー消費効率

(II) JIS B 8616

表 8 JIS B 8616 試験条件

		室内側吸込空気温度		室外側吸込空気温度	
		乾球温度	湿球温度	乾球温度	湿球温度
定格 条件	標準気候	27℃	19℃	35℃	24℃
	低温気候	21℃	15℃	27℃	19℃
	高温気候	29℃	19℃	46℃	24℃
低温条件		27℃	19℃	29℃	19℃

・本体表示事項

定格冷房能力[kW]：全熱能力（製造業者が指定する定格冷房能力の95%以上）
定格冷房消費電力[kW]（製造業者が指定する定格冷房消費電力の105%以下）

・技術資料公表事項（技術資料などに適切に表示）

定格冷房時の顕熱比（SHF）
冷房エネルギー消費効率
中間冷房能力[kW]（定格冷房能力の50%±5%）
中間冷房消費電力[kW]
中間冷房エネルギー消費効率
冷房期間エネルギー消費効率

(III) JIS C 9617

表 9 JIS C 9617 試験条件

試験条件		乾球温度	湿球温度
標準 (定格)		27℃	21.2℃
過負荷		32℃	23℃
低温	A	18℃	13.5℃
	B	5.0℃	2.1℃

定格除湿能力[L/日] (製造業者が指定する定格除湿能力の 95%以上)

定格消費電力[W] (製造業者が指定する定格消費電力の 110%以下)

(IV) 住宅向け商品カタログデータ

表 10 住宅向け商品カタログデータ

	室内側吸込空気温度		室外側吸込空気温度		表示項目
	乾球温度	相対湿度	乾球温度	相対湿度	
再熱除湿 (JRA 除湿条件)	24℃	60%	24℃	80%	消費電力[W] 除湿量[L/h] 吹出し温度[℃]
再熱除湿 (P 社条件)	27℃	60%	27℃	80%	除湿量[L/h] 吹出し温度[℃]
弱冷除湿 (D 社条件)	28℃	60%	28℃	—	除湿量[cc/h] 吹出し温度[℃] 顕熱能力[W] 消費電力[W]
除湿機	27℃	60%	—	—	除湿量[L/日] 消費電力[W]

現在、住宅向け空調機については潜熱処理能力について JIS 規格で必要表示項目として定められていない。一方、一体型の室内循環型の除湿専用機（再熱除湿機構）については、定格条件における除湿性能の表示を定められている。また、空調機の再熱除湿運転モードについては JEMA の規格で性能評価点が定められている。但し、性能表示されている空調設備システムにおいても室内温湿度条件は一点のみの表記となっており、実運転状態における性能を評価するには十分な情報とは言えない。

(V) 業務用商品設備設計資料データ

表 11 業務用空調機設備設計資料データ

	室内側吸込空気温度		室外側吸込空気温度		表示項目
	乾球温度	湿球温度	乾球温度	相対湿度	
空調機	27℃	19℃	35℃	—	定格能力[kW] 風量[m ³ /min] 電動機出力[W] SHF (顕熱比) 消費電力[W] バイパスファクター
外調機	—	—	33℃	28℃	定格能力[kW] 風量[m ³ /h] 電動機出力[W] SHF (顕熱比) 消費電力[W] バイパスファクター
全熱交 併設外調機	27℃	19℃	35℃	24℃	温度交換効率 エンタルピー交換効率 定格冷房能力[kW] 全熱交分冷房能力[kW] 運転電流[A] 消費電力[W] 電動機出力[W]
除湿機	25℃	80%	—	—	除湿量[L/h] 消費電力[kW] 運転電流[A] 力率[%] 風量[m ³ /min] 除湿機能力線図
除湿機	30℃	60%	—	—	除湿量[L/day] 定格消費電力[kW] 始動電流[A] 風量[m ³ /min] ファン電動機出力[kW] 除湿機能力線図
除湿機	27℃	60%	—	—	除湿量[L/day] 定格消費電力[kW] 始動電流[A] 風量[m ³ /min] ファン電動機出力[kW] 除湿機能力線図

一方、業務用空調機については、設備設計資料上で空調機の SHF（顕熱比）を表示することが JIS 規格上要求されており、設備設計資料上で定格能力、SHF、バイパスファクターが表記されている。しかし、室内温湿度条件は定格条件の一点のみとなっており、室内温湿度の変化や負荷率の変化による影響はデータからは予測できないため実運転状態における性能を評価するためには十分ではないと考えられる。一方、業務用（産業用）の除湿機（再熱除湿機能）については定格条件における性能だけでなく除湿機能力線図が与えられており、室内温湿度変化時の能力が予測できる。ただし、消費電力の変化についての情報が記載されていないため実運転時の消費電力量予測が困難であることに加え、効率が低く一般空調設備としての利用は難しい。

3.1.1.2 吸着除湿を行う空調設備システム

(1) 動作原理

水を吸着する材料（吸着材）は相対湿度の変化によって吸着できる水の量が変化する。この状態を示す線図を吸着等温線と呼ぶ。下図に代表的な吸着材の吸着等温線を示す。

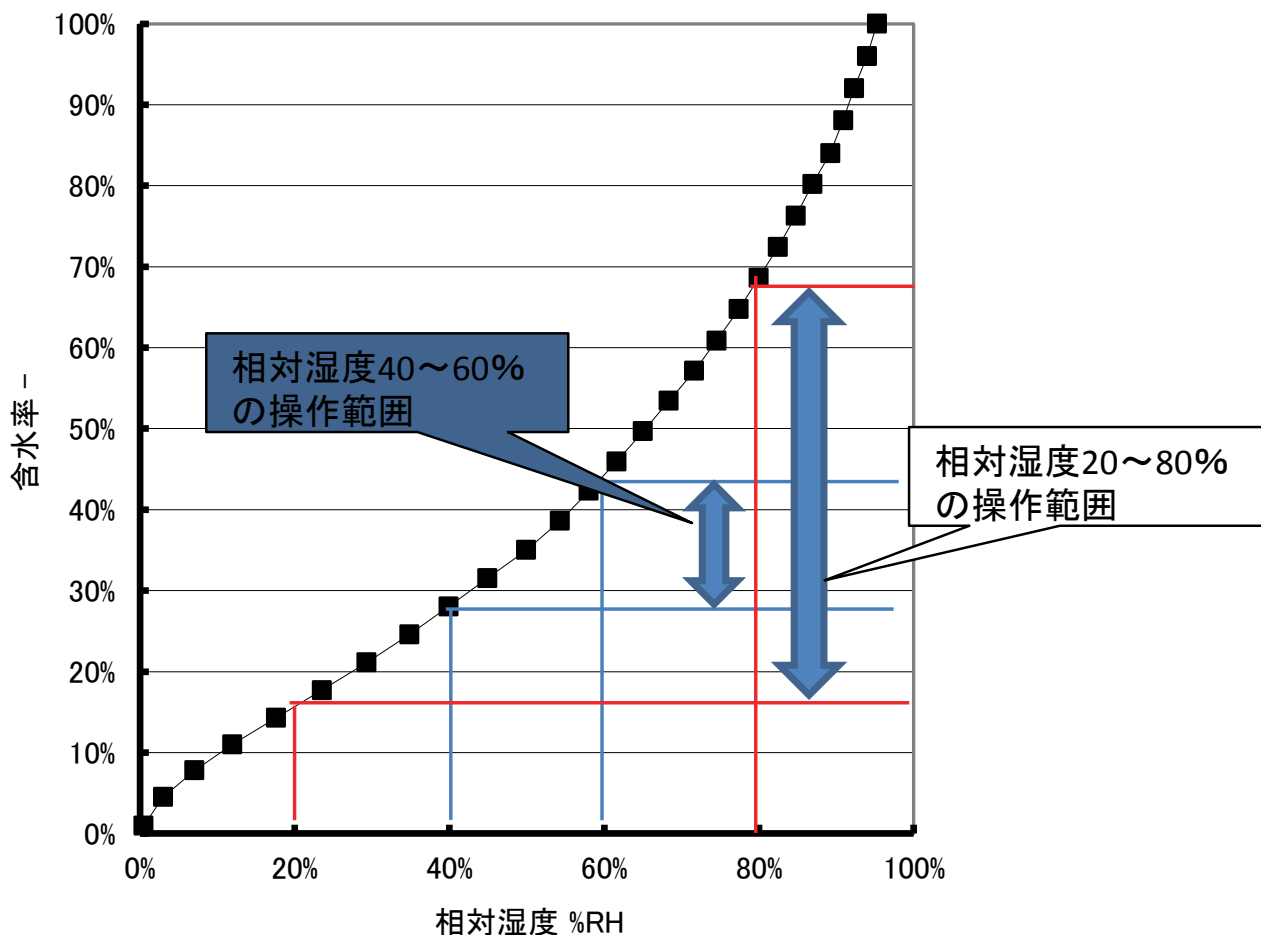


図 28 代表的な吸着材料の吸着等温線

吸着等温線の縦軸である含水率は、吸着した水の質量／吸着材の質量で表すことが出来る。図 29 に示すように吸着材を流通する空気の相対湿度を変化させる際に水分が吸着、脱着する現象を利用して除湿運転を行う。図 28 に示すように、相対湿度の変化幅を大きく取ると吸着材に水を吸放湿できる量が増大するため少ない吸着材の量で高い除湿性能を得ることが出来る。

この、吸着現象を用いた除湿システムの代表例であるデシカント空調機の模式図と空気線図を下图に示す。

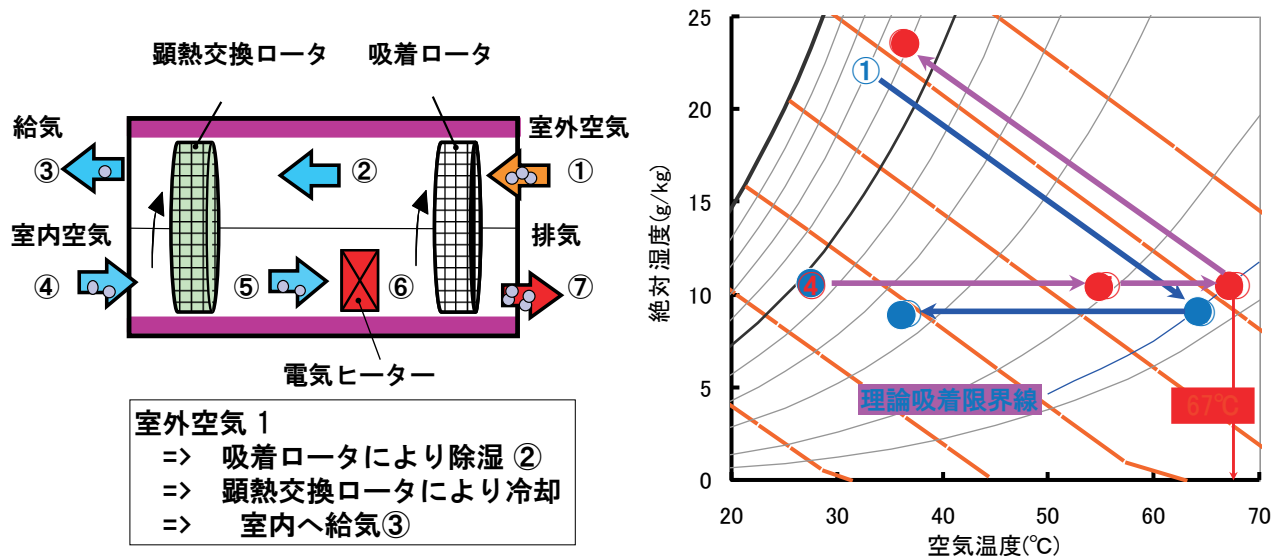


図 29 デシカント空調機の模式図と空気線図

図 29 に示すように、暖かい湿った空気が吸着ローターを流通する際に水分を吸着すると同時に吸着熱を発生し、熱い乾いた空気を吹き出す。熱い乾いた空気を室内空気と熱交換して冷却して室内に流通する。吸着ローターは、室内空気を加熱した空気を流通して乾燥することによって吸着した水を脱着して再生される。

この時、再生をする空気は動作原理上少なくとも吸着ローターで、水を吸着させた後の熱い乾いた空気よりも低い相対湿度になるまで加熱する必要がある。上記に示すように 33°C、22 g/kg の空気を 9 g/kg まで除湿するためには最低でも 67°C の加熱温度が必要となる。実際には、放熱ロス、ヒーターと空気の熱伝達ロス、空気とローター間の熱伝達ロス、ローターの熱容量ロス、吸着熱≧凝縮熱であることから約 100°C の加熱が必要となる。従って、高圧蒸気やヒーターなどの比較的効率低い熱源が必要となることから効率が低く一般空調には採用されてこなかった。しかし、近年これらの課題を克服することを目的とした種々の方式が開発されている。

(2) 吸着除湿を行う空調設備システム

(I) 住宅向けハイブリッド除湿機

再熱除湿とデシカント除湿を組み合わせたシステム。一度冷却除湿で除湿された空気をデシカントで吸着除湿することにより、吸着部分での除湿量を低減することによって吸着熱の発生を抑制し室内温度が低温条件以外ではヒーター加熱なしで十分な除湿量を確保可能とする。

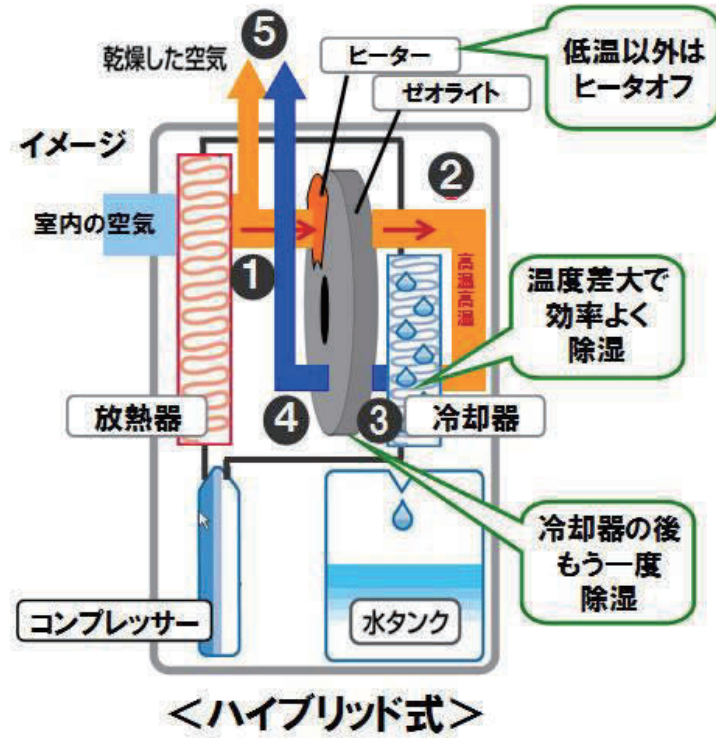


図 30 住宅向けハイブリッド除湿機

(II) 業務用冷却除湿ハイブリッドシステム

住宅向け除湿機と同様に、業務用空調機にも冷却除湿と吸着除湿を組み合わせたシステムが存在する。代表的なシステムの模式図と空気線図を下図に示す。

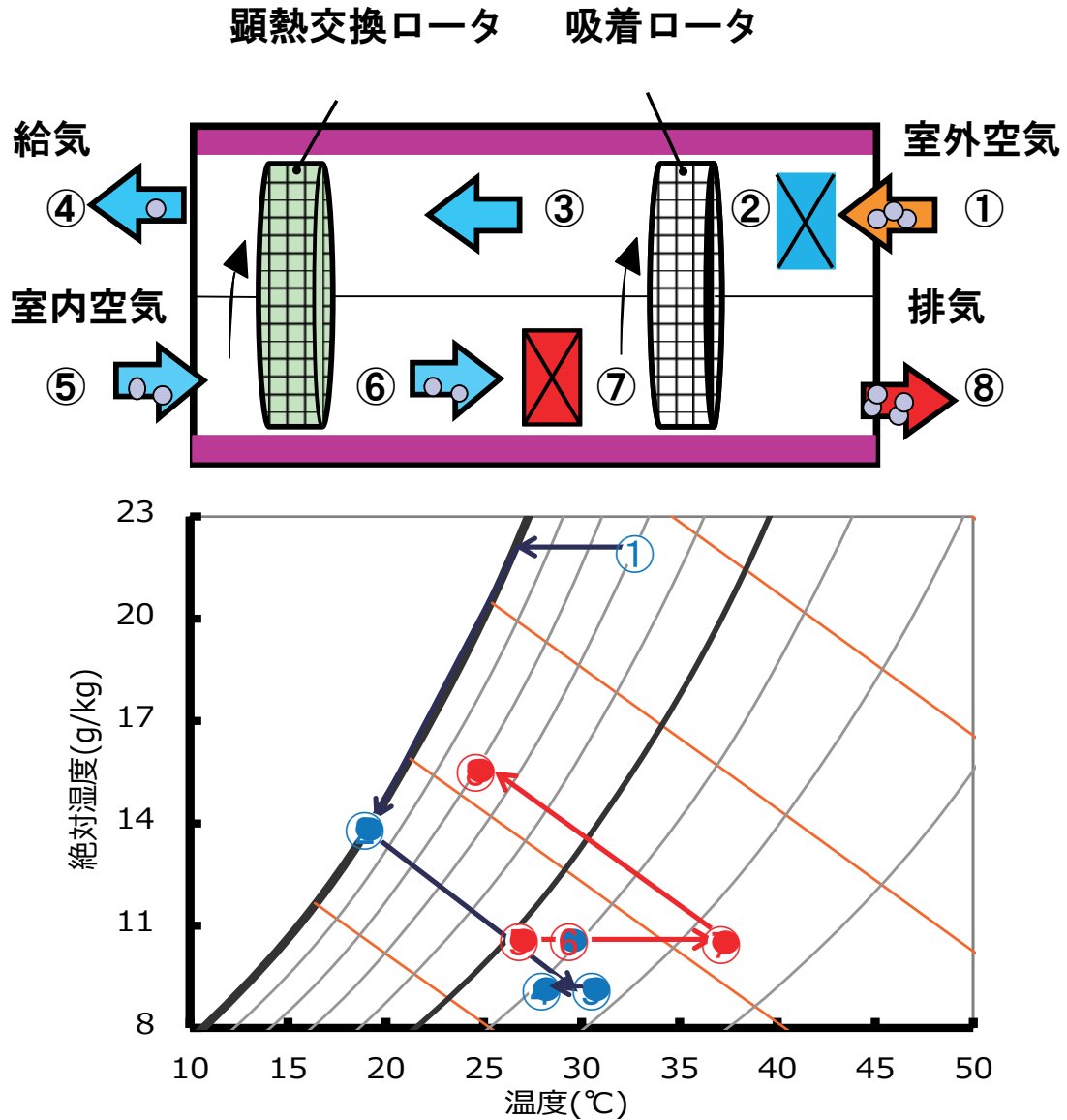


図 31 冷却除湿ハイブリッドシステム

図 31 に示すように、住宅向けのハイブリッドシステムと同様に吸着ローターに流通する前に冷却除湿によって除湿を行うことによって吸着ローターにおける吸着熱の発生を抑制している。これによって、低温再生を可能とし、ヒートポンプの凝縮熱程度の熱量でも運転可能なシステムを実現している。

(III) 吸着ローター式全熱交換器

吸着現象を利用した全熱交換器システムが存在する。代表的なシステムの模式図と空気線図を下図に示す。

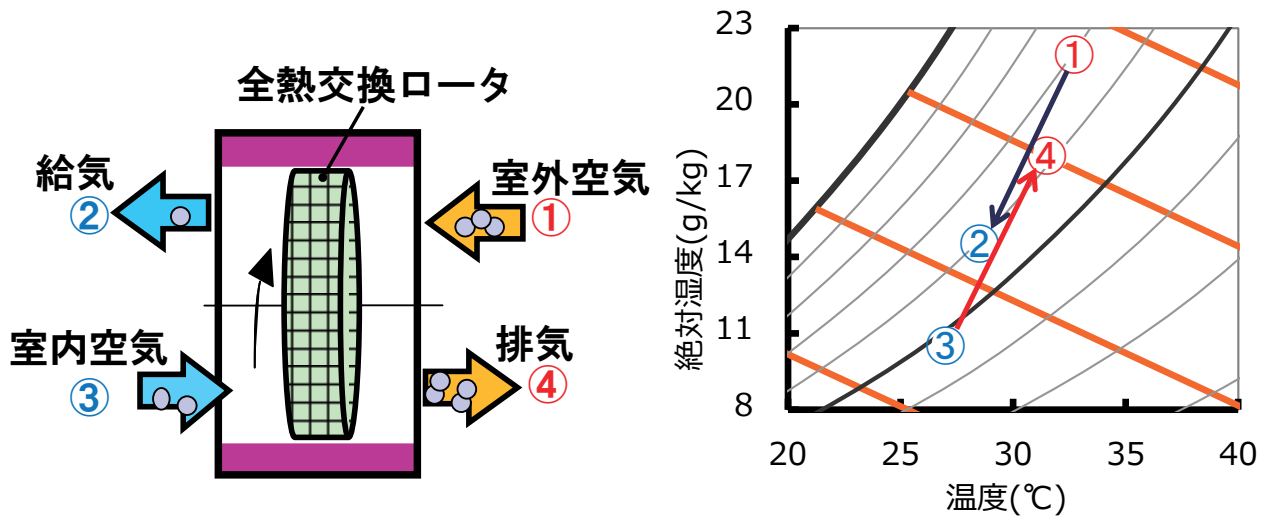


図 32 吸着ローターを用いた全熱交換器

図 32 に示すように、室外空気が吸着ローターを流通する際にローターに熱を蓄熱するとともに水を蓄湿し、室内空気側流通側で放湿する。これにより、室内空気と室外空気の間で顕熱と同時に潜熱を交換する。静止型のプレートフィン式全熱交換器と比較して一般的に吸着ローターを用いた全熱交換器は潜熱交換効率が低い。

(IV) 直接冷却吸着、直接加熱脱着方式

吸着熱発生によって再生温度が高くなってしまふことを回避するために、吸着熱を直接除熱しながら吸着を行うシステム。代表的なシステムの模式図と空気線図を下図に示す。

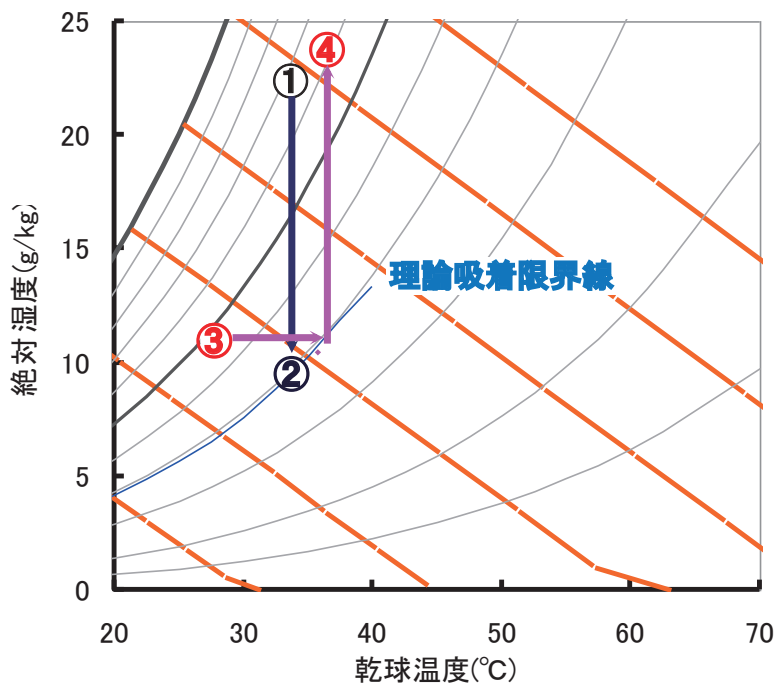
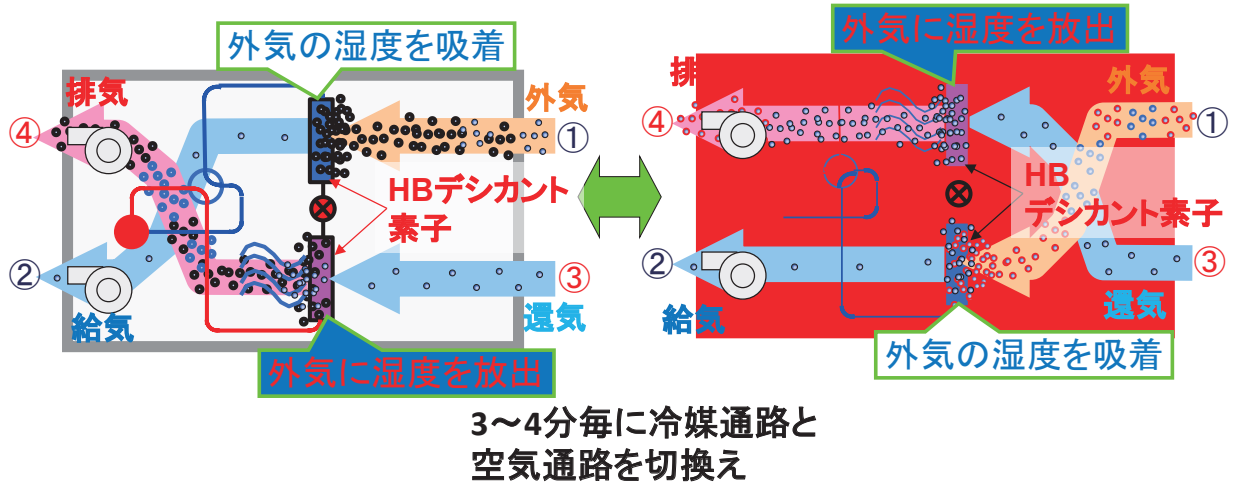


図 33 直接冷却吸着、直接加熱脱着システム

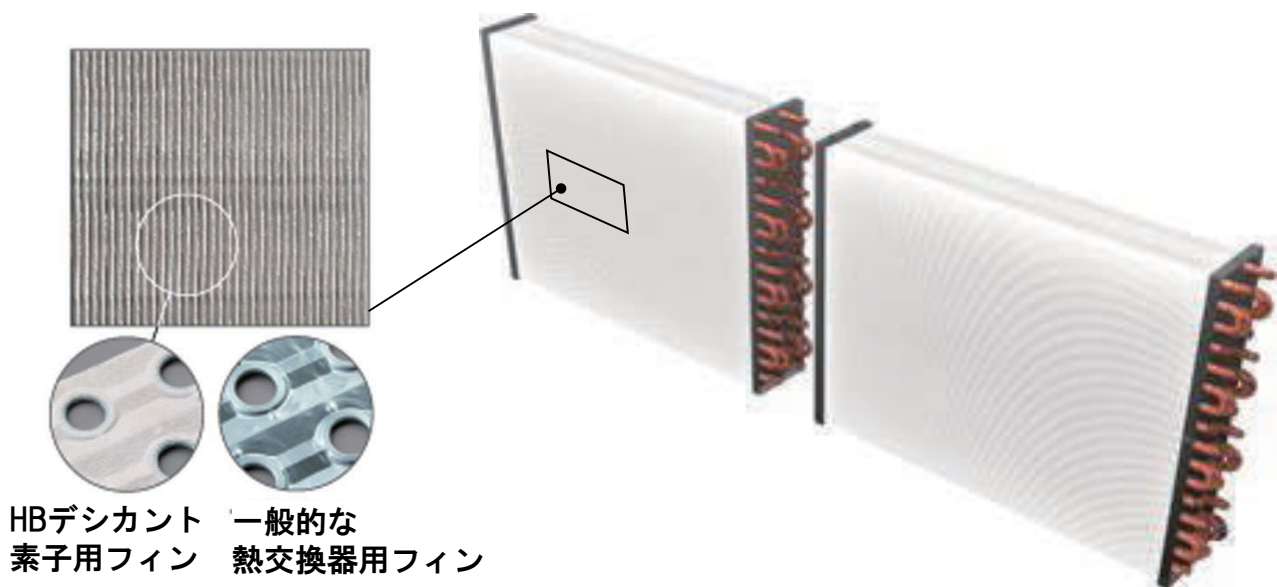


図 34 HB デシカント素子

図 34 に示す通常の空調機に用いられるクロスフィン熱交換器の上に、吸着材をコーティングした HB デシカント素子の伝熱管内に冷媒を流通することによって水を吸着する際の吸着熱を直接除熱、水を脱着する際の脱着熱を直接加熱する。

これによって、図 33 の空気線図に示すように、水を脱着するために必要な再生温度を大幅に低減している。

図 33 の模式図に示すように二つの HB デシカント素子を搭載し一定時間毎に冷媒回路と空気通路を切り換えるバッチ運転を行うことによって連続的な除湿運転を実現している。

(3) 吸着除湿を行う空調設備システムの規格で定められた性能評価点、カタログ、技術資料性能評価点

吸着除湿を行う空調設備システムは、吸着型の全熱交換器のみ JIS で定められた規格が存在する。

表 12 吸着除湿を行う空調設備システムの規格一覧

番号	名称	適用範囲
JISB8628	全熱交換器	居住空間などの快適な空気調和における省エネルギーを目的とした空気対空気の全熱交換器について規定する

(I) JIS B 8628

表 13 JIS B 8628 試験条件

室内側吸込空気温度		室外側吸込空気温度		表示項目
乾球温度	湿球温度	乾球温度	湿球温度	
27℃	20℃	35℃	29℃	風量（定格風量の±10%以内） 静圧損失（仕様書表示値の±10%以内）または機外静圧 熱交換効率（仕様書表示値の90%以上） ・温度交換効率 ・エンタルピー交換効率 消費電力または電動機の定格出力 ・有効換気量 小型（250m ³ /h 未満）：定格風量の85%以上 中型（250m ³ /h 以上 2000m ³ /h 以下） ：定格風量の90%以上 大型（2000m ³ /h 超える）：定格風量の85%以上

(II) 住宅向け商品カタログデータ

表 14 住宅向け商品カタログデータ

	室内側吸込空気温度		室外側吸込空気温度		表示項目
	乾球温度	相対湿度	乾球温度	相対湿度	
ハイブリッド方式	27℃	60%	—	—	消費電力[W] 除湿量[L/日]
デシカント方式	20℃	60%	—	—	消費電力[W] 除湿量[L/日]

(III) 業務用商品設備設計資料データ

表 15 業務用空調機設備設計資料データ

	室内側吸込空気温度		室外側吸込空気温度		表示項目
	乾球温度	相対湿度 湿球温度	乾球温度	湿球温度	
除湿機	15℃	40%	—	—	定格能力[L/h] 風量[m ³ /min] 定格消費電力[W] 除湿機能力線図
全熱交換器	—	—	—	—	機種選定表風量比 風量と熱回収率[%]の関係
外調機 直接冷却吸着 直接加熱脱着	27℃	19℃	33℃	28℃	全熱能力[kW] 顕熱能力[kW] 消費電力[W] 風量[m ³ /h] 除湿最大性能特性表

吸着ローターを用いた空調設備システムは大型の設備が多く、受注生産品が多いため（客先要望に併せて性能を決める）性能特性が公開されていないものが多い。

全熱交換器は、比較的多くのデータが開示されているが熱回収率が潜熱と顕熱で分離されていないため潜熱処理能力を予測するためのデータとしては不十分である。

直接冷却吸着、直接加熱脱着を用いた外調機は比較的多くのデータが開示されているが負荷率が変化した際の性能と消費電力については記載されていないため実運転状態の性能を技術資料の開示データだけで推測することは難しい。一方、公共的なエネルギー計算手法である国交省の LCEM ツールの特性式に性能予測が可能なデータ（特性式）が提供されている。

3.1.1.3 膜透過除湿を行う空調設備システム

(1) 動作原理

膜透過除湿の動作原理を図 35 に示す。

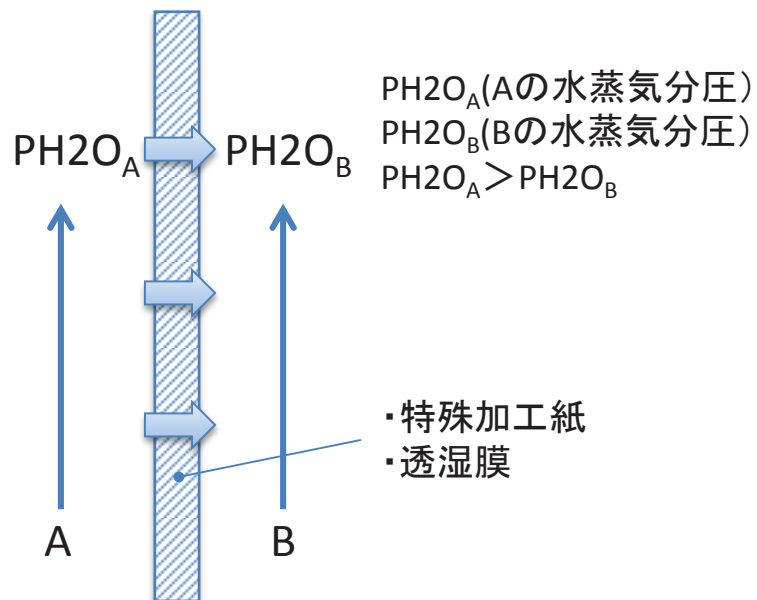


図 35 膜透過除湿の原理

図 35 に示すように、水を選択的に通す特殊加工紙や高分子透湿膜を介して二つの流体を流通し、二つの流体の蒸気圧差によって湿度を移動させて除湿運転を行う。

(2) 膜透過除湿を行う空調設備システム

(I) プレートフィン式全熱交換器

膜透過除湿を行う空調設備システムの代表例として、換気を行いながら室内空気と室外空気との顕熱と潜熱を同時に熱交換するプレートフィン式全熱交換器が存在する。

模式図と空気線図を図 36 に示す。

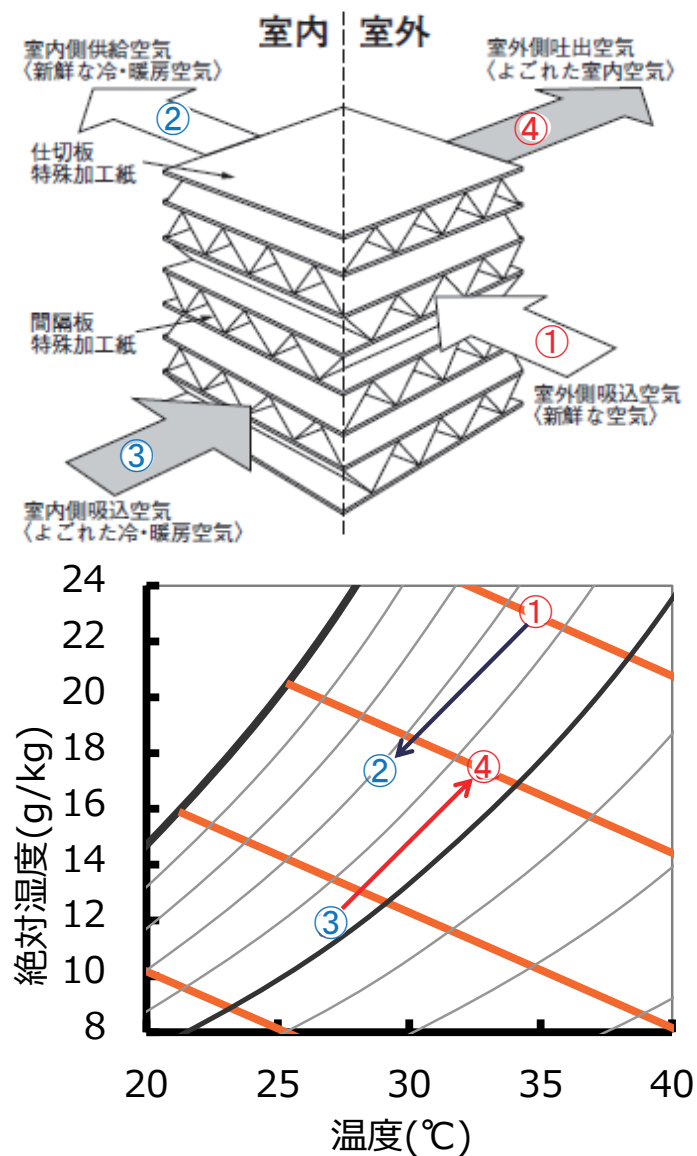


図 36 プレートフィン式全熱交換器

プレートフィン式全熱交換器は、室外空気の給気と室内空気の排気の間で熱と同時に湿度を熱交換して熱回収運転を行う。給気と排気は、透湿性を有する仕切り板で仕切られており、給気と排気を混合することなく換気運転を実施する。

3.1.1.4 気化式加湿器の状態変化

- ・ 気化式は、水を加湿材に浸透させて気化蒸発させる。
- ・ 本変化は、水は気化蒸発して絶対湿度を高め、空気は水との熱伝達及び水の蒸発潜熱によって冷やされる。潜熱は増えて、顕熱は減少する。
- ・ 実用上、潜熱と顕熱の授受は等しい(エンタルピーの増減がない断熱変化)と考えて差し支えなく、空気線図上は以下の図のように湿球温度一定の線上(エンタルピー一定の線上と考えても差し支えない)を変化する。

気化式加湿・水噴霧加湿の空気線図上の変化

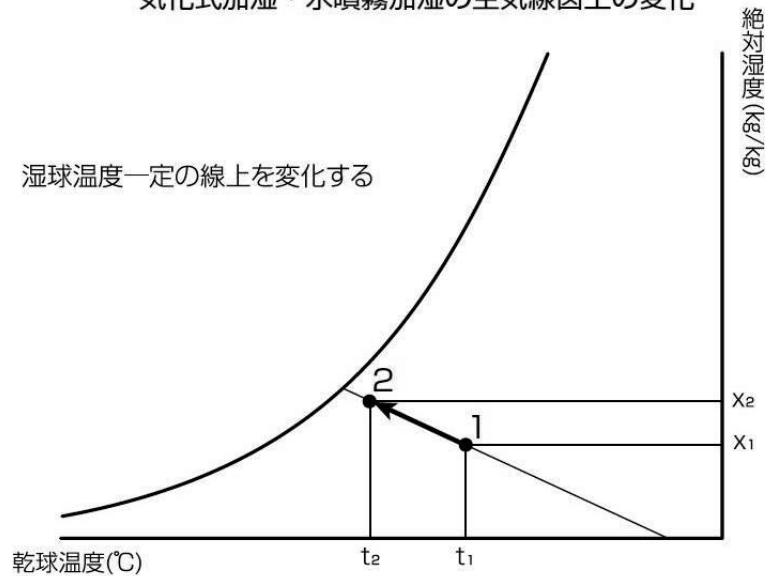


図 37 空気線図上の変化

飽和効率について

- ・ 飽和効率は加湿による空気の状態変化の中で、飽和点に至るまで、どこまで加湿できるか(加湿のしやすさ)を示す。
- ・ 図 38 に気化式加湿器の空気線図を記す。①の空気状態から湿球温度一定の線上を②まで加湿したとする。次に②の状態点から延長していくと飽和点③に至る。この② - ①、③ - ①の線分比が飽和効率である。

【気化式加湿器の飽和効率】

$$\text{飽和効率} = \frac{\text{②} - \text{①}}{\text{③} - \text{①}} \times 100(\%) = \frac{X_2 - X_1}{X_3 - X_1} \times 100(\%) = \frac{t_2 - t_1}{t_3 - t_1} \times 100(\%)$$

式 6

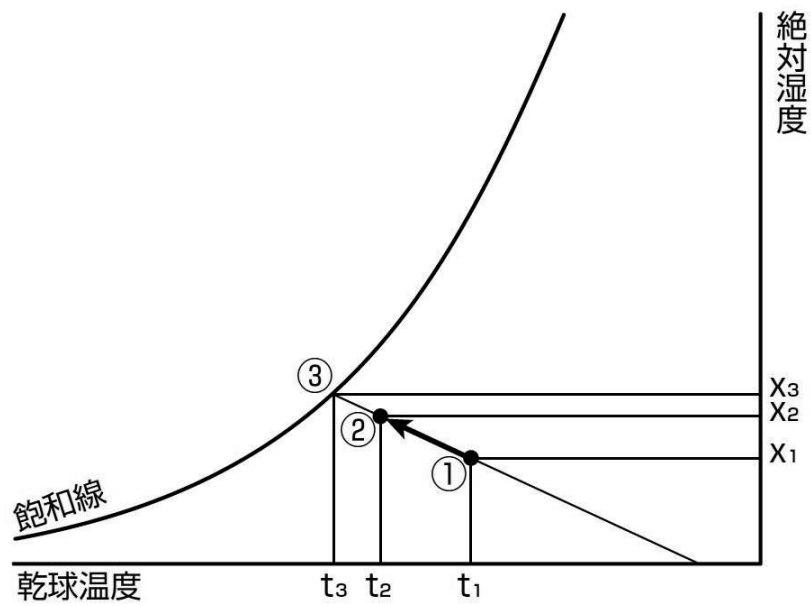


図 38 空気線図上の変化

また、以下に実際に記されているカタログ飽和効率を示す。機器ごとに数値が異なり、飽和効率が 1～80%までレンジがある。よって、空気条件、目標湿度によって、機種選定を行う必要がある。(参照：ウエットマスター株式会社 VDC シリーズ)

表 16 飽和効率とモジュール奥行寸法の関係

	製品 A-1	製品 A-2	製品 A-3	製品 A-4
加湿モジュール 奥行寸法[mm]	50	65	100	130
飽和効率[%]	1～45	46～55	56～70	71～80

3.1.2 潜熱負荷処理メカニズムを考慮したエネルギー消費量ロジック

これまで種々の原理、規格について記した。以下に、実際に用いられる空調システムを夏期と冬期に分けて記す。

3.1.2.1 <夏期>

(1) 空調システム：パッケージエアコン+換気装置の場合



図 39 空調システム：パッケージエアコン+換気装置

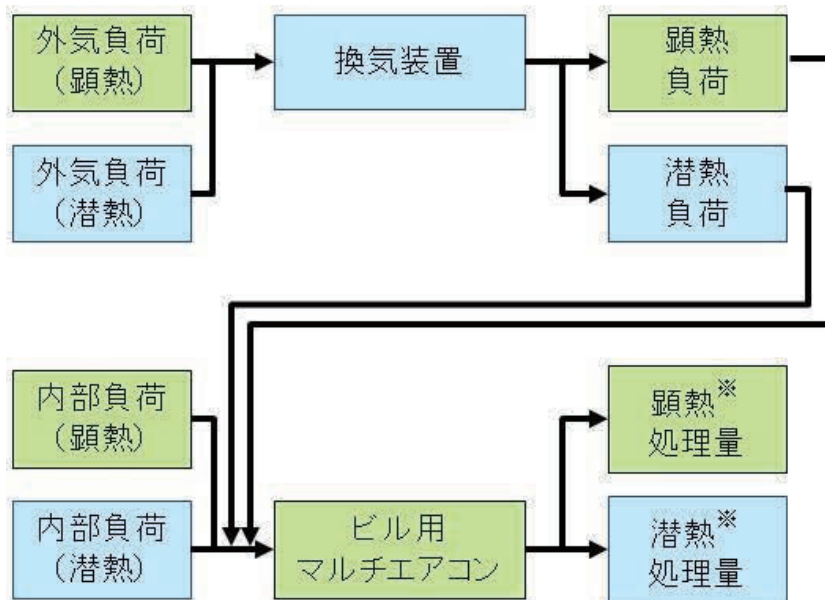


図 40 負荷処理ロジック

換気装置は外気負荷（顕熱・潜熱）を処理する機械ではない為、負荷がそのまま室内に流入する。よって、ビル用マルチエアコンは内部負荷（顕熱・潜熱）と外気負荷（顕熱・潜熱）を処理する。ただし、負荷条件によって SHF は決まるのでビル用マルチエアコンは顕熱処理もしくは、潜熱処理のどちらかを目標とした運転を実施する事になる。

(2) 空調システム：パッケージエアコン+全熱交換器の場合



図 41 空調システム：パッケージエアコン+全熱交換器

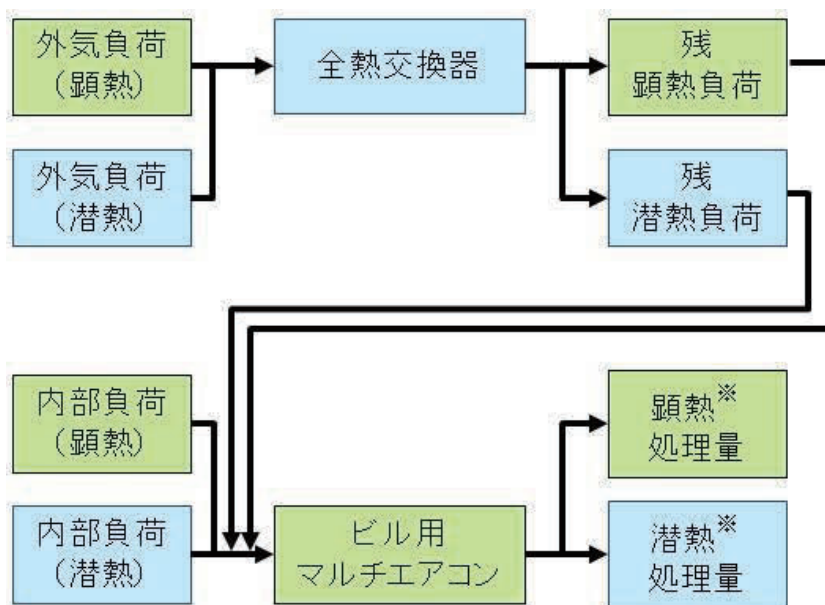


図 42 負荷処理ロジック

全熱交換器は換気によって失われる空調エネルギーの顕熱、潜熱を交換回収する省エネルギー装置である。具体的には、室内から排出する顕熱・潜熱を全熱交換器内部の元素にて外気と熱交換する事により、給気時に負荷を低減させている。残顕熱負荷をビル用マルチエアコンによって処理を行う。ただし、残負荷条件によって SHF は決まるのでビル用マルチエアコンは顕熱処理もしくは、潜熱処理のどちらかを目標とした運転を実施する事になる。

(3) 空調システム：パッケージエアコン+調湿外気処理機の場合



図 43 空調システム：パッケージエアコン+調湿外気処理機

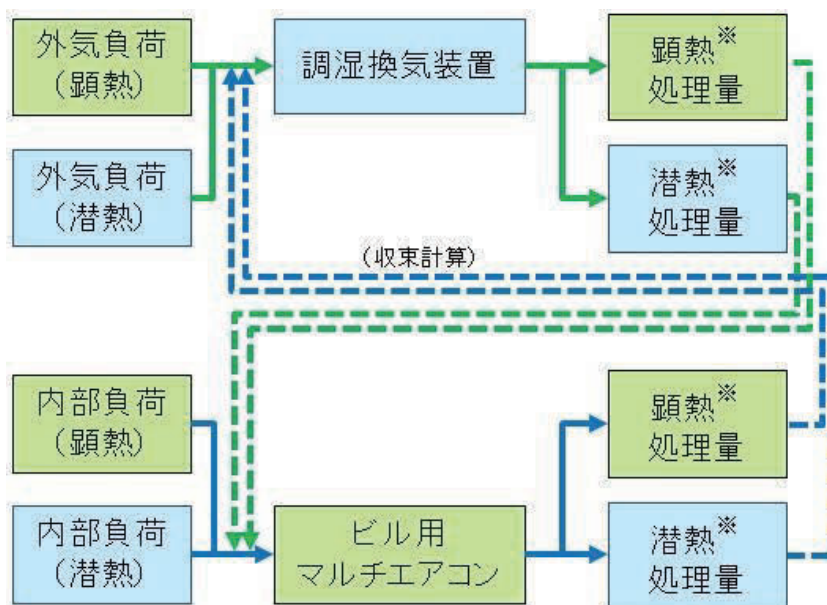


図 44 負荷処理ロジック

高顕熱形ビル用マルチエアコンおよびヒートポンプデシカント調湿外調機の負荷率を変化させて収束計算を行う事で、外気負荷（顕熱・潜熱）、内部負荷（顕熱・潜熱）を全て処理する。※を記した顕熱処理量、潜熱処理量は、負荷率によって、処理量が変更する。

3.1.2.2 <冬期>

(1) 空調システム：パッケージエアコン+換気装置の場合



図 45 空調システム：パッケージエアコン+換気装置

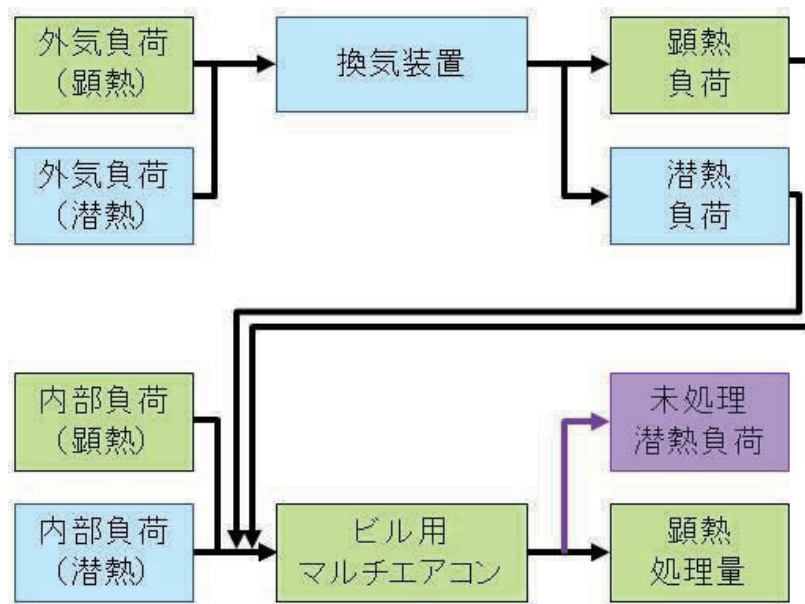


図 46 負荷処理ロジック

換気装置は外気負荷（顕熱・潜熱）を処理する機械ではない為、負荷がそのまま室内に流入する。ただし、ビル用マルチエアコンは潜熱負荷を処理しない為、顕熱負荷のみを処理し、潜熱負荷は未処理となる。

(2) 空調システム：パッケージエアコン+全熱交換器の場合



図 47 空調システム：パッケージエアコン+全熱交換器

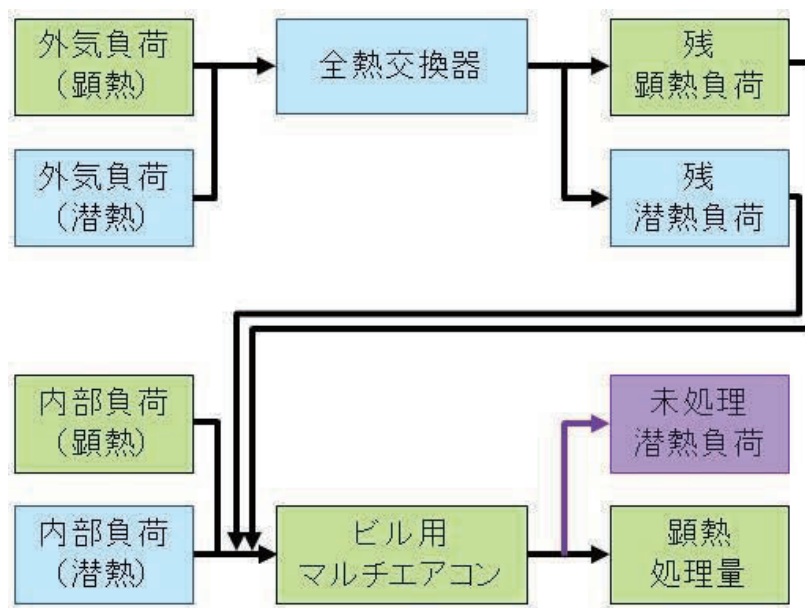


図 48 負荷処理ロジック

全熱交換器によって、室内空気と外気を熱交換する事により、給気時に負荷を低減する。残顕熱負荷をビル用マルチエアコンにて処理を行う。ただし、ビル用マルチエアコンは潜熱負荷を処理しない為、潜熱負荷は未処理となる。

(3) 空調システム：パッケージエアコン+換気装置+気化式加湿器の場合



図 49 空調システム：パッケージエアコン+換気装置+気化式加湿器

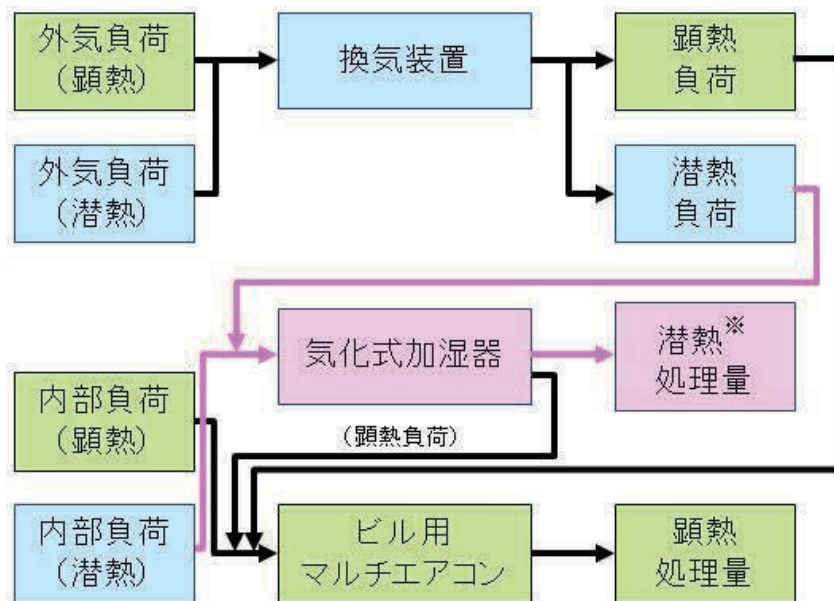


図 50 負荷処理ロジック

気化式加湿器を用いる事で、外気と内部由来の潜熱負荷を処理する。ただし、加湿器の飽和効率によっては目標湿度未達の場合もある。尚、空気中に水分を放出し気化した際に、周辺空気の熱を奪うので、顕熱負荷が発生する。顕熱負荷はビル用マルチエアコンが処理を行う。

(4) 空調システム：パッケージエアコン+全熱交換器+気化式加湿器の場合



図 51 空調システム：パッケージエアコン+全熱交換器+気化式加湿器

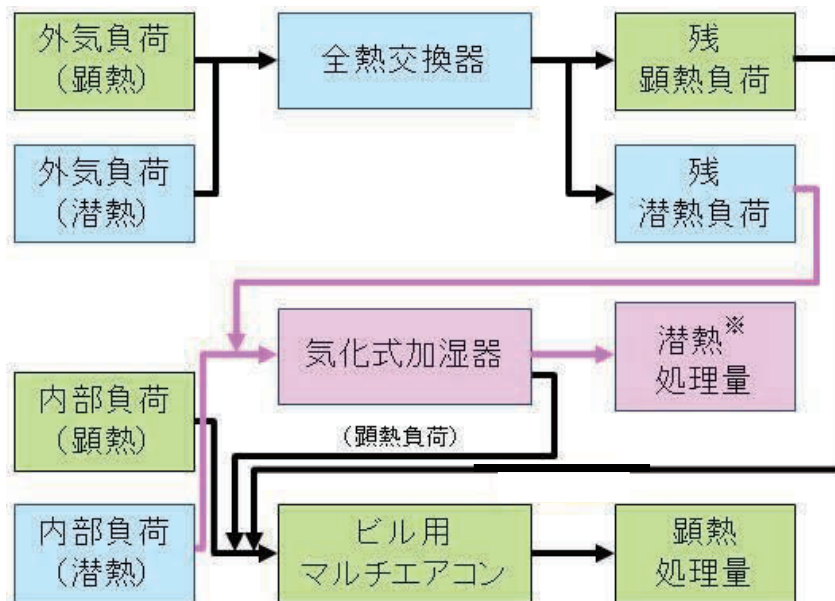


図 52 負荷処理ロジック

気化式加湿器を用いる事で、外気と内部由来の潜熱負荷を処理する。ただし、加湿器の飽和効率によっては目標湿度未達の場合もある。尚、空気中に水分を放出し気化した際に、周辺空気の熱を奪うので、顕熱負荷が発生する。顕熱負荷はビル用マルチエアコンが処理を行う。

(5) 空調システム：パッケージエアコン+全熱交換器（気化式加湿器内蔵）の場合



図 53 空調システム：パッケージエアコン+全熱交換器（気化式加湿器内蔵）

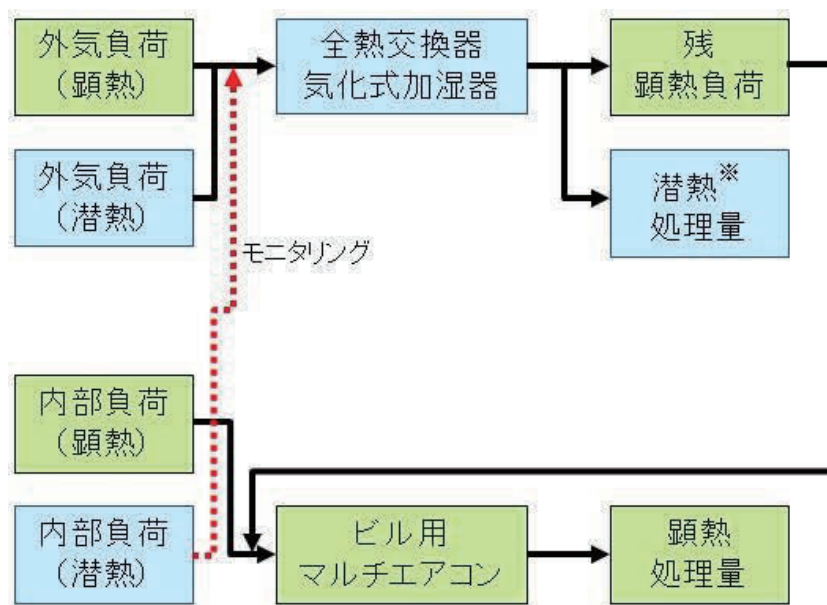


図 54 負荷処理ロジック

全熱交換器（気化式加湿器内蔵）にて潜熱負荷を処理する。その為、室内側潜熱負荷を還気吸込み時にモニタリングを行い、外気潜熱負荷と内部潜熱負荷を併せて処理を行う。ただし、飽和効率によっては、残潜熱負荷が発生する。

(6) 空調システム：パッケージエアコン+調湿外気処理機の場合



図 55 空調システム：パッケージエアコン+調湿外気処理機

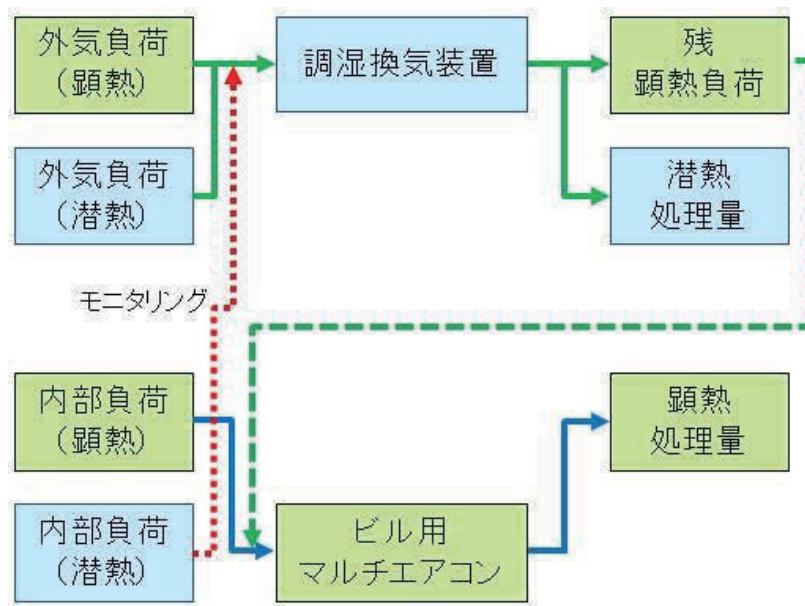


図 56 負荷処理ロジック

ヒートポンプデシカント調湿外調機にて外部由来の潜熱負荷と内部由来の潜熱負荷を処理する。その為、室内側潜熱負荷を還気吸込み時にモニタリングを行い、外気潜熱負荷と内部潜熱負荷を併せて処理を行う。高顕熱ビル用マルチエアコンにて残顕熱を処理する。

3.1.3 対象建築物、空調システム概要

3.1.3.1 東京 A ビル

建築物概要

東京 A ビルは都内某所に建築された業務用建築物である。表 17 に建築概要を記す。

表 17 東京 A ビル建築概要

用途	事務所
竣工	2013 年
延床面積	7157.18m ² (2,165.04 坪)
基準階面積	625.41m ² (189.18 坪)
構造	鉄骨造 (一部鉄骨鉄筋コンクリート造)
規模	地上 9 階、地下 1 階
基準階天井高	2800mm

実測は、2 階で実施した。平面図を図 57 に記す。

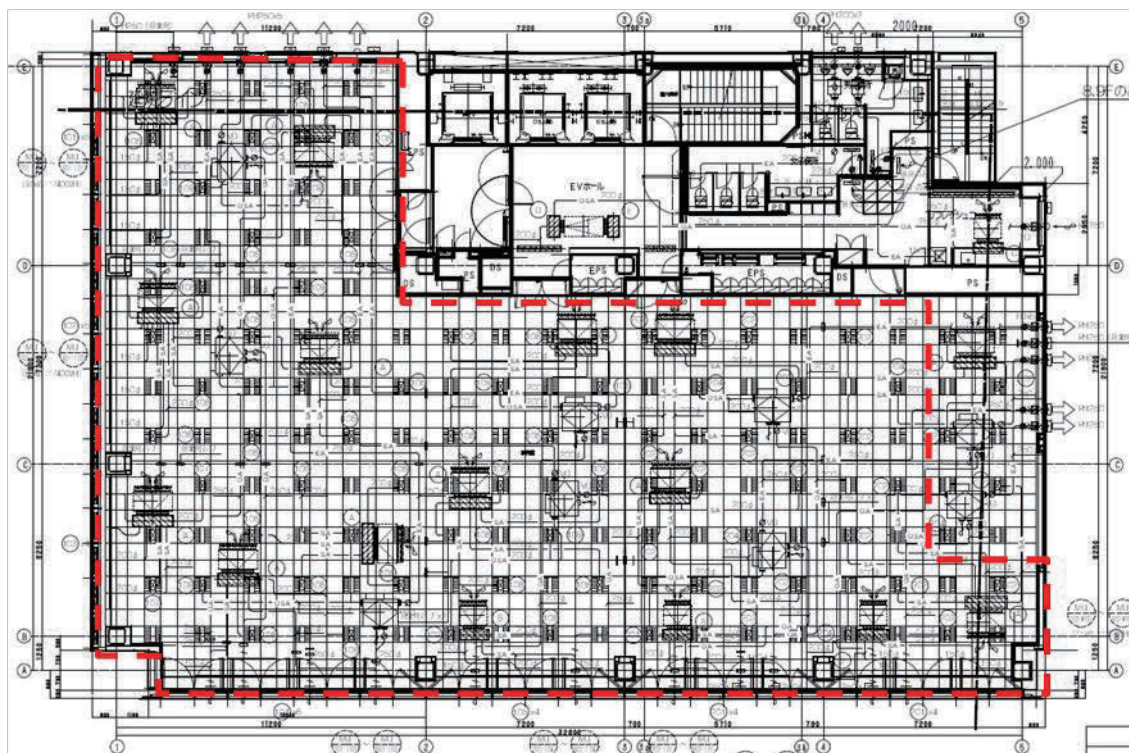


図 57 基準階 (2F) 平面図

赤点線で示した部分が執務室スペースとなっており、その他部分はエレベーター、エレベーターホール、トイレ、給湯室、PS、EPS、その他の共用部スペースとなっている。尚、執務室に入室する為の、ドアはセキュリティーカードを用いて入室の必要があり、基本的にドアが閉められている。また、執務者は最大 100 人程度が在室している。

本ビルは、「ダブルスキンカーテンウォール」を採用しており、南面にカーテンウォールを設置して、夏場は太陽光を遮蔽している。また、冬期に関しては導入する外気を暖めて、室内の換気装置に導入している。図 58 にイメージを記す。

図 59 にダブルスキンカーテンウォールと接続している拡大図面を記す。(冬期：ダブルスキンカーテンウォールにて外気が暖められる状態イメージ)

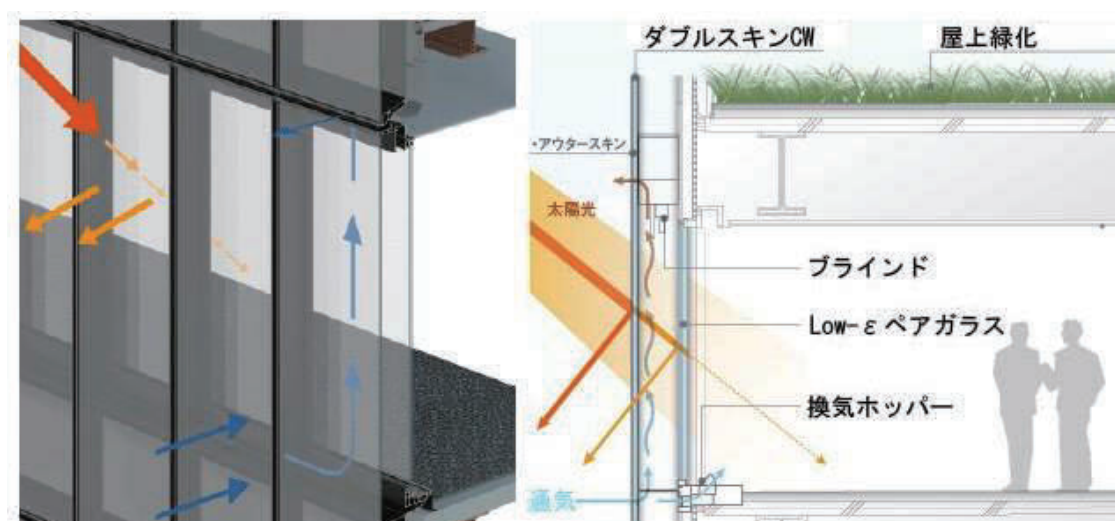


図 58 ダブルスキンカーテンウォール

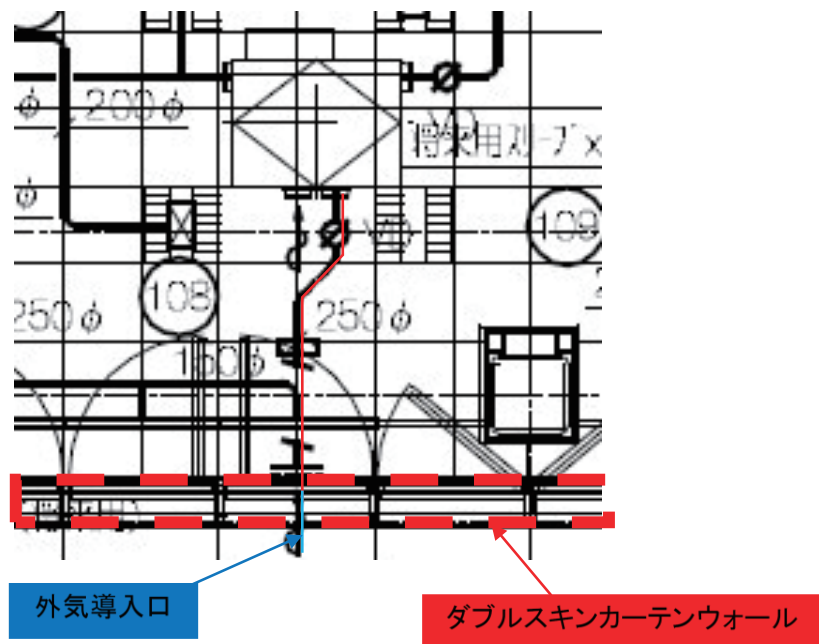


図 59 ダブルスキンカーテンウォール拡大図（図面下部が南面）

(1) 空調システム

図 60 に室内機の配置を記す。東京 A ビルの空調設備は天井隠蔽ダクト式になっており、接続するフレキシブルダクトとアネモによって室内へ送風を実施している。吸込みは天井内空間に設置されており、天井空間をチャンバーとして用いている。（天井チャンバー方式）

室内機設置位置を赤枠で記す。執務室用に延べ 16 台の室内機を設置している。本研究においては、二重赤枠で示す、3 台の室内機（P01、P05、I04）の能力を測定した。尚、3 台の室内機は同一熱源（室外機）に接続されている。表 18、19 に室外機と室内機の機器仕様を記す。同一室外機に接続される室内機は、FXYMP140AA×4、FXYMP90AA×5、FXYMP71AA×1 の 10 台が接続されている（計測を実施する 3 台を含む）。

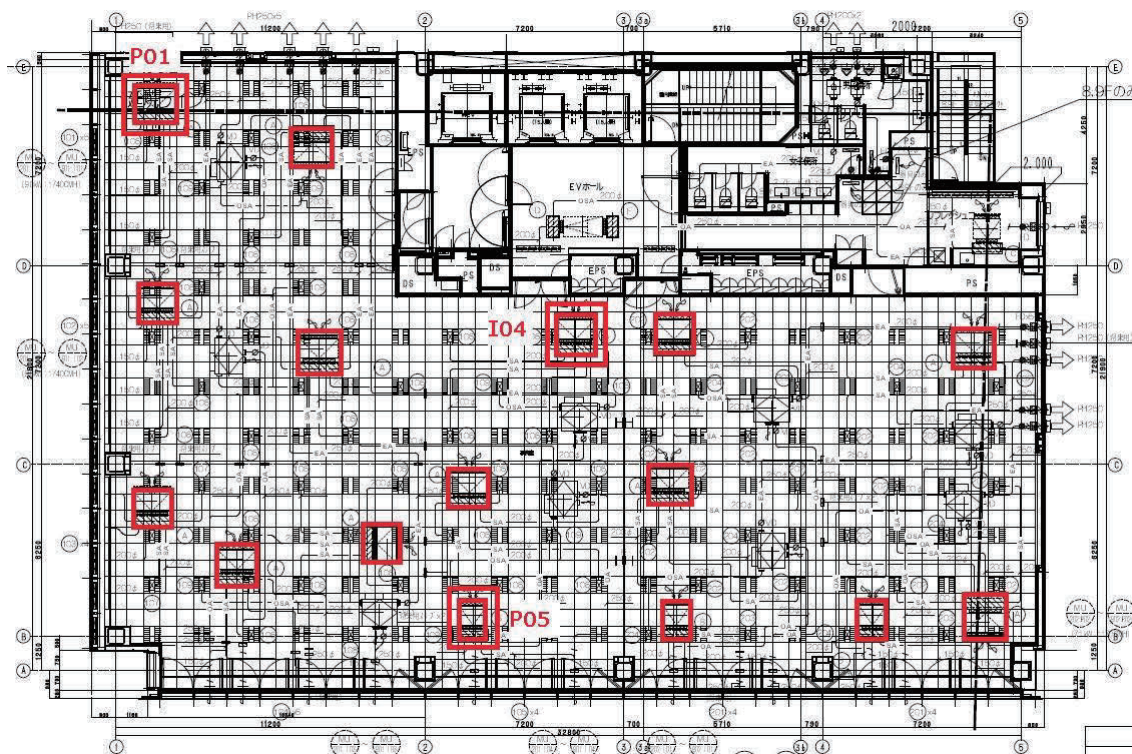


図 60 東京 A ビル空調機設置位置

表 18 熱源（室外機）仕様

室外機（形式）：マルチパッケージ形空気調和機（冷暖房同時形高頭熱形）	
機器名称	ビル用マルチ Ve-upⅢR
型番	REKP800AA
機器仕様	定格冷暖房能力（冷房/暖房）：80.0 / 90.0kW 定格消費電力（冷房/暖房）：24.8 / 24.9kW 室外ユニット COP：（冷房/暖房）3.23 / 3.61 定格試験条件：JIS B 8615 冷房_室内 27°CDB19°CWB、外気 35°CDB JIS B 8615 暖房_室内 20°CDB、外気 7°CDB,6°CWB ※各室内ユニットまでの配管相当長 7.5m、高低差 0m、ラウンドフロー室内ユニット 100%接続時の値 圧縮機電動機出力：3.3+4.5+2.7+4.5+4.5kW,ファン電動機出力：0.75+0.35+0.35kW 電源：三相 200V 冷媒配管：液側 φ 19.1、吸入ガス側 φ 31.8、高低圧ガス側 φ 28.6、均圧側 φ 19.1mm

表 19 室内機仕様

室内ユニット：P01	
機器名称	天井埋込ダクト形
型番	FXYPMP140AA
機器仕様	定格冷暖房能力：14.0 / 16.0kW 定格消費電力（冷房/暖房）：0.279 / 0.267kW 定格試験条件：JIS B 8616 冷房：室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房：室内 20°CDB、外気 7°CDB,6°CWB 定格機外静圧：100Pa ファン電動機出力：0.35kW 電源：単相 200V 冷媒配管：液側φ9.5、ガス側φ15.9mm ファン風量（急-強-弱）：39-33-28m ³ /min
室内ユニット：P05	
機器名称	天井埋込ダクト形
型番	FXYPMP71AA
機器仕様	定格冷暖房能力：7.1 / 8.0kW 定格消費電力（冷房/暖房）：0.134 / 0.122kW 定格試験条件：JIS B 8616 冷房_室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房_室内 20°CDB、外気 7°CDB,6°CWB 定格機外静圧：100Pa ファン電動機出力：0.35kW 電源：単相 200V 冷媒配管：液側φ9.5、ガス側φ15.9mm ファン風量（急-強-弱）：19.5-17.5-16m ³ /min
室内ユニット：I04	
機器名称	天井埋込ダクト形
型番	FXYPMP90AA
機器仕様	定格冷暖房能力：9.0 / 10.0kW 定格消費電力（冷房/暖房）：0.184 / 0.172kW 定格試験条件：JIS B 8616 冷房_室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房_室内 20°CDB、外気 7°CDB,6°CWB 定格機外静圧：100Pa ファン電動機出力：0.35kW 電源：単相 200V 冷媒配管：液側φ9.5、ガス側φ15.9mm ファン風量（急-強-弱）：29-25-22m ³ /min

(2) 外調システム

図 61 東京 A ビル外調機設置位置に外調機の配置を記す。(設置位置を青枠) 外調機も接続するフレキシブルダクトとアネモによって、室内へ送風を実施している。吸込みは室内機同様に天井内チャンバー内に設置されている。執務室用に述べ 9 台の外調機を設置している。ただし、右端のバツ印で示す 2 台の外調機については、執務室の都合上常時停止にて運用している。

本調査においては、二重青枠で示す、2 台の換気装置 (D02、D04) の能力を測定した。表 20 外調機仕様に換気装置の詳細仕様を記す。

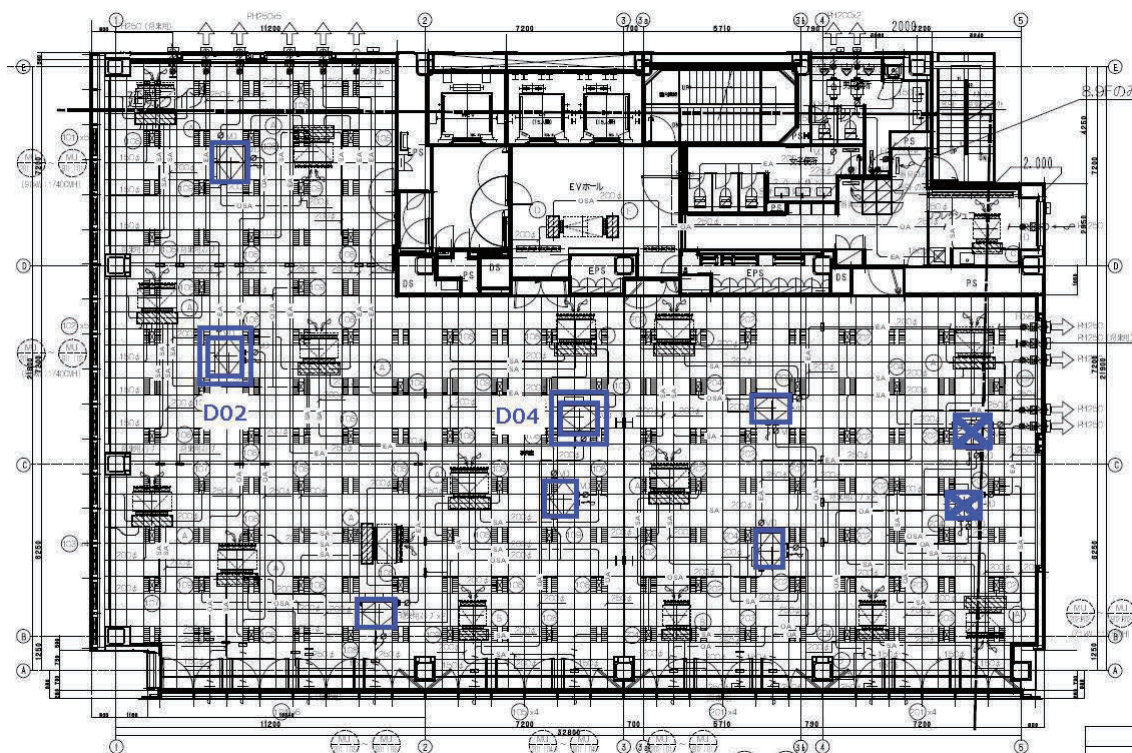


図 61 東京 A ビル外調機設置位置

表 20 外調機仕様

水配管レス調湿外気処理器：D02	
機器名称	ヒートポンプデシカント調湿外調機
型式	HDMP50C
機器仕様	定格除湿冷房能力：5.5kW（全熱）、除湿量 6.67kg/h 定格加湿暖房能力：7.1kW（全熱）、加湿量 3.78kg/h 定格試験条件：冷房除湿 室内 27°CDB,19°CWB、外気 33°CDB,28°CWB 暖房加湿 室内 22°CDB,15.5°CWB、外気 0°CDB,-3°CWB 定格消費電力（冷房除湿/暖房加湿）：1.1 / 1.54kW 定格風量：500m ³ /h ファン電動機出力：140×2W 電源：単相 200V
水配管レス調湿外気処理器：D04	
機器名称	ヒートポンプデシカント調湿外調機
型式	HDMP25C
機器仕様	定格除湿冷房能力：2.8kW（全熱）、除湿量 3.33kg/h 定格加湿暖房能力：3.5kW（全熱）、加湿量 1.89kg/h 定格試験条件：冷房除湿 室内 27°CDB,19°CWB、外気 33°CDB,28°CWB 暖房加湿 室内 22°CDB,15.5°CWB、外気 0°CDB,-3°CWB 定格消費電力（冷房除湿/暖房加湿）：0.68 / 0.79kW 定格風量：250m ³ /h ファン電動機出力：(53×2) W 電源：単相 200V

外調機は、HDMP50C×6、HDMP25C×3 の 9 台が設置されている。外調機は、それぞれが冷媒配管で接続されているわけではなく、それぞれが独立で運転している。ただし、2 階に設置されている外調機は 2 台（HDMP50C×1、HDMP25C×1）が常時運用を停止している。

3.1.3.2 東京 B ビル

建築物概要

東京 B ビルは都内某所に建築された業務用建築物である。表 21 に建築概要、図 62 に平面図を示す。実測は、2 階で実施した。

表 21 東京 B ビル建築概要

用途	事務所
竣工	2013 年
延床面積	7,430.02m ² (2,247.58 坪)
基準階面積	642.91 m ² (194.48 坪)
構造	鉄骨造 (一部鉄骨鉄筋コンクリート造)
規模	地上 9 階、地下 1 階
基準階天井高	2800mm

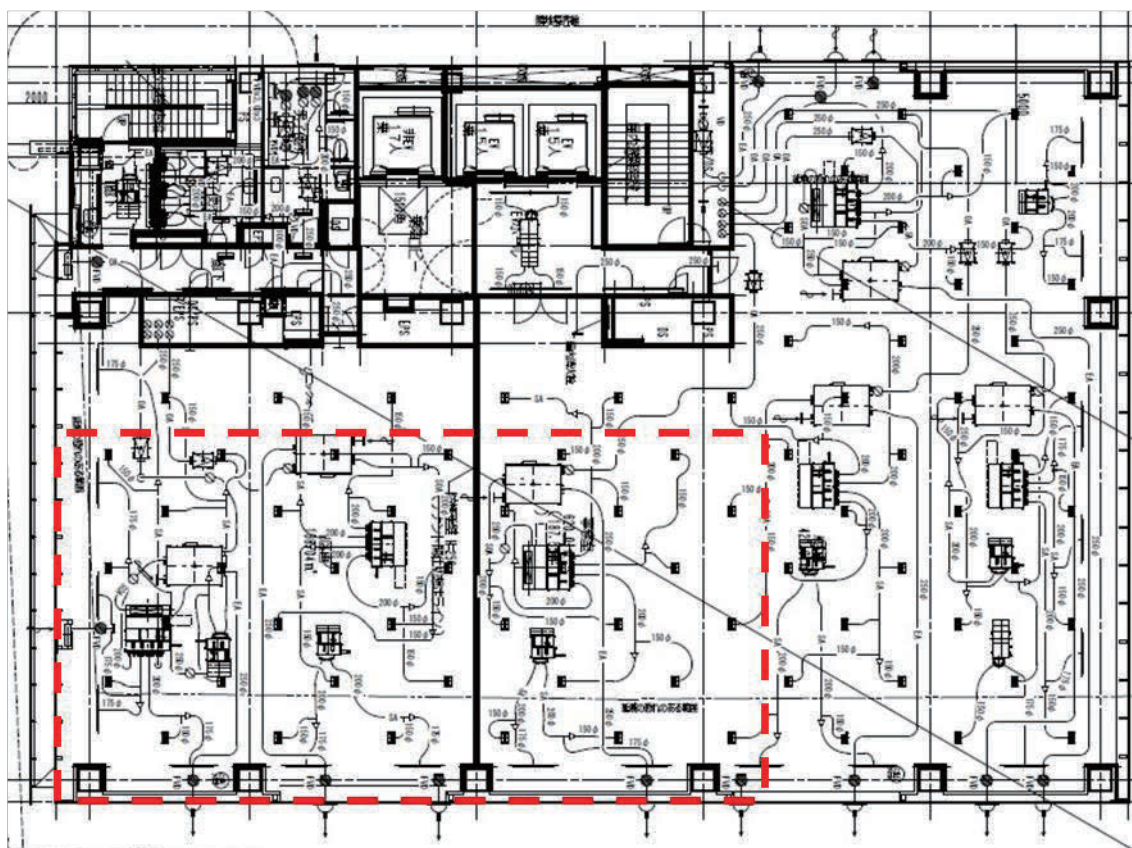


図 62 基準階 (2F) 平面図

赤点線で示した部分が執務室スペースとなっており (壁で区切られている)、その他部分はエレベーターホール、会議室、トイレ、給湯室などの共用部スペースとなっている。尚、執務室は基本的にドアが閉められている。また、執務者は最大で 65 人程度が在室している。

表 23 室内機仕様

室内ユニット：P01	
機器名称	天井埋込ダクト形
型番	FXYP45AA
機器仕様	定格冷暖房能力（冷房/暖房）：4.5 / 5.0kW 定格消費電力（冷房/暖房）：0.146 / 0.134kW 定格試験条件：JIS B 8616 冷房 室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房 室内 20°CDB、外気 7°CDB,6°CWB 定格機外静圧：100Pa ファン電動機出力：0.14kW 電源：単相 200V 冷媒配管：液側φ6.4、ガス側φ12.7 ファン風量（急・強・弱）：16-13-11（m ³ /min）
室内ユニット：P05	
機器名称	天井埋込ダクト形
型番	FXYP112AA
機器仕様	定格冷暖房能力（冷房/暖房）：11.2 / 12.5kW 定格消費電力（冷房/暖房）：0.210 / 0.198kW 定格試験条件：JIS B 8616 冷房_室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房_室内 20°CDB、外気 7°CDB,6°CWB 定格機外静圧：100Pa ファン電動機出力：0.35kW 電源：単相 200V 冷媒配管：液側φ9.5、ガス側φ15.9 ファン風量（急・強・弱）：32-27-23（m ³ /min）

上記に示すシステムは、計測を実施している室外機、室内機を示す。同一室外機に接続される室内機は、FXYP45AA×2、FXYP56AA×1、FXYP112AA×1、FXYP140AA×1、換気装置VKMP100GAM×2の7台が接続されている（計測を実施する2台を含む）。

(2) 外調システム

図 64 に外調機の配置を記す。(設置位置を青枠) 外調機も接続するフレキシブルダクトとアネモによって、室内へ送風を実施している。吸込みは室内機同様に天井内チャンパー内に設置されている。2階には述べ6台の換気装置を設置しており、測定対象の執務室用に3台の換気装置を設置している。

本研究においては、二重青枠で示す、1台の換気装置(D01)の能力を測定した。表 24 に詳細仕様を記す。

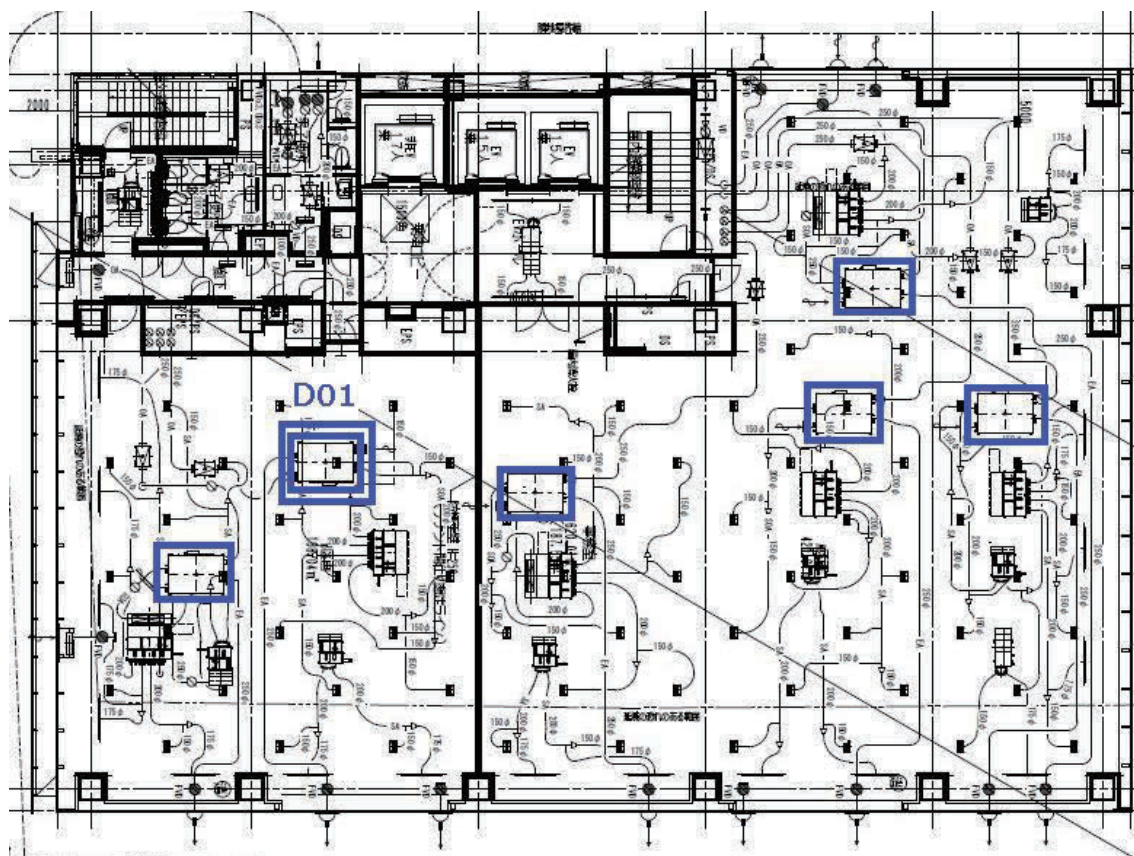


図 64 東京 B ビル外調機設置位置

表 24 外調機仕様

外気処理タイプ 室内ユニット : D01	
機器名称	全熱交換器 (加湿器標準装備、冷却暖房コイル付)
型式	VKMP100GAM
機器仕様	定格冷房能力 : 9.12kW、 定格暖房能力 : 10.69kW 定格試験条件 : JIS B 8616 冷房_室内 27°CDB、19°CWB、外気 35°CDB JIS B 8616 暖房_室内 20°CDB、外気 7°CDB、6°CWB 消費電力 (特強-強-弱) : 0.556-0.506-0.418kW 風量 (特強-強-弱) : 950-950-820m ³ /h ファン電動機出力 : (0.28×2) kW 機外静圧 (特強-強-弱) : 110-75-70 温度交換効率 (特強-強-弱) : 74-74-76.5% エンタルピー交換効率_冷房 (特強-強-弱) : 62-62-66 エンタルピー交換効率_暖房 (特強-強-弱) : 65-65-69 加湿量 : 5.4kg/h 電源 : 単相 200V 冷媒配管 : 液側 φ 6.4、吸入ガス側 φ 12.7、給水配管 : φ 6.4 ドレン配管 : φ PT3/4 オネジ

前述した通り、換気装置も室外機 REYPP615AA に接続されており、対象執務室に 3 台設置している。(計測を実施する 1 台を含む)。

図 65 に全熱交換器の機内イメージ図を記す (ダイキン工業カタログ抜粋)。

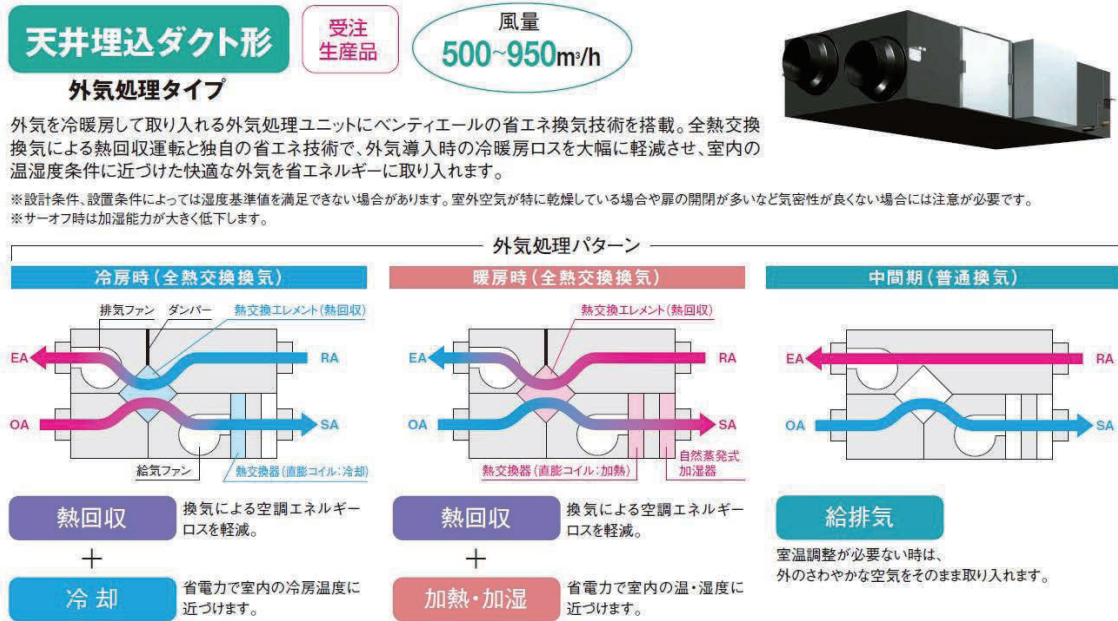


図 65 全熱交換器機内イメージ

VKMP100GAM は、冷房時：熱回収+冷却運転、暖房時：熱回収+加熱、加湿運転、中間期：給排気運転を自動で選択し、運転を実施する。

3.1.3.3 大阪 C ビル

建築物概要

大阪ビルは大阪府内某所に建築された業務用建築物であり、表 25 に概要を、図 66 に平面図を記す。

表 25 大阪 C ビル建築概要

用途	事務所
基準階面積	721.5m ²
規模	地上 3 階、地下 1 階
基準階天井高	2800mm

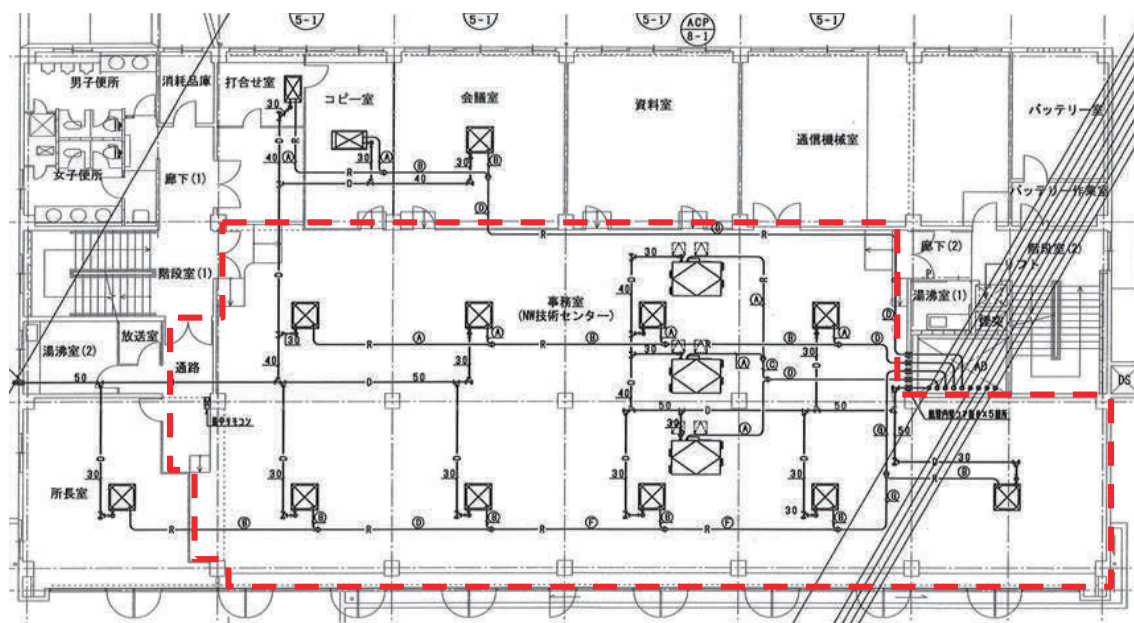


図 66 基準階 (2F) 平面図

赤点線で示した部分が執務室スペースとなっており、その他部分は階段、廊下、会議室、トイレなどの共用部スペースとなっている。尚、執務室に入室する為の、ドアはセキュリティーカードを用いて入室の必要があり、基本的にドアが閉められている。また、執務者は最大で 50 人程度が在室出来る。

(1) 外調システム

図 66 に外調機の配置を記す。(設置位置を青枠) 外調機は接続するフレキシブルダクトとアネモによって、室内へ送風を実施している。吸込みは東京 A ビル、B ビル同様に天井内チャンバー内に設置されている。2 階には延べ 3 台の換気装置を設置しており、測定対象の執務室用に 3 台の換気装置を設置している。

本研究においては、二重青枠で示す、3 台の換気装置 (D01、D02、D03、) の能力を測定した。

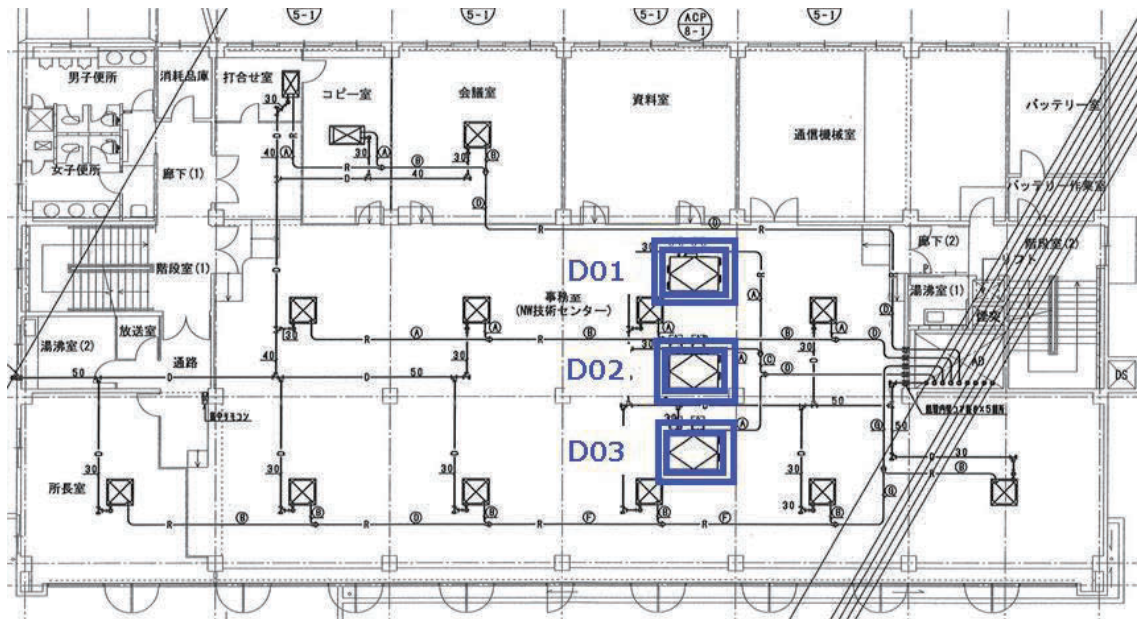


図 67 大阪 C ビル外調機設置位置

表 26 に外調機の機器仕様を記す。

表 26 外調機機器仕様

外気処理タイプ室内ユニット：D01、D02、D03	
機器名称	全熱交換器（加湿器標準装備、冷却暖房コイル付）
型式	VKMP100GAM
機器仕様	定格冷房能力：9.12kW、 定格暖房能力：10.69kW 定格試験条件：JIS B 8616 冷房_室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房_室内 20°CDB、外気 7°CDB,6°CWB 消費電力（特強-強-弱）：0.556-0.506-0.418kW 風量（特強-強-弱）：950-950-820m ³ /h ファン電動機出力：(0.28×2) kW 機外静圧（特強-強-弱）：110-75-70 温度交換効率（特強-強-弱）：74-74-76.5% エンタルピー交換効率_冷房（特強-強-弱）：62-62-66 エンタルピー交換効率_暖房（特強-強-弱）：65-65-69 加湿量：5.4kg/h 電源：単相 200V 冷媒配管：液側_φ6.4、吸入ガス側_φ12.7、 給水配管：φ6.4 ドレン配管：φPT3/4 オネジ

換気装置は室外機 RXUP218BA に接続されており、対象執務室に 3 台設置している。

3.1.3.4 国立研究開発法人 建築研究所 環境実験棟

建築物概要

国立研究開発法人建築研究所環境実験棟は茨城県つくば市に建築された実験用建築物である。表 27 に建築概要、図 68 に平面図を示し、図 69 に屋上平面図を記す。測定は、4 階で実施した。平成 27 年 2 月 9 日～平成 27 年 2 月 23 日：一期工事、平成 27 年 7 月 6 日～平成 27 年 7 月 31 日：二期工事を実施して以下の空調機、換気装置を据え付けた。対象の室は模擬オフィスとされ、様々な試験を実施できるようにしている。本調査で取り付けした空調機、換気装置を図 68、図 69 に据付図面、表 28～表 34 にて据付空調機仕様と計測機器について記す。

表 27 環境実験棟建築概要

用途	模擬オフィス
基準階面積	72 m ² (21.7 坪)
規模	地上 4 階
基準階天井高	2700mm

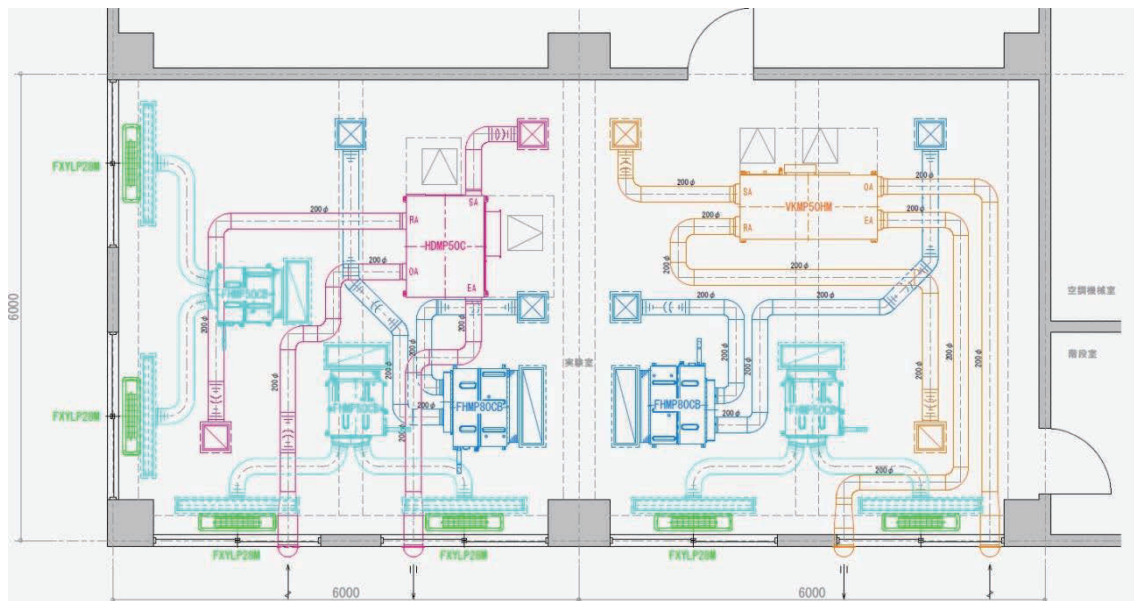


図 68 基準階 (2F) 平面図

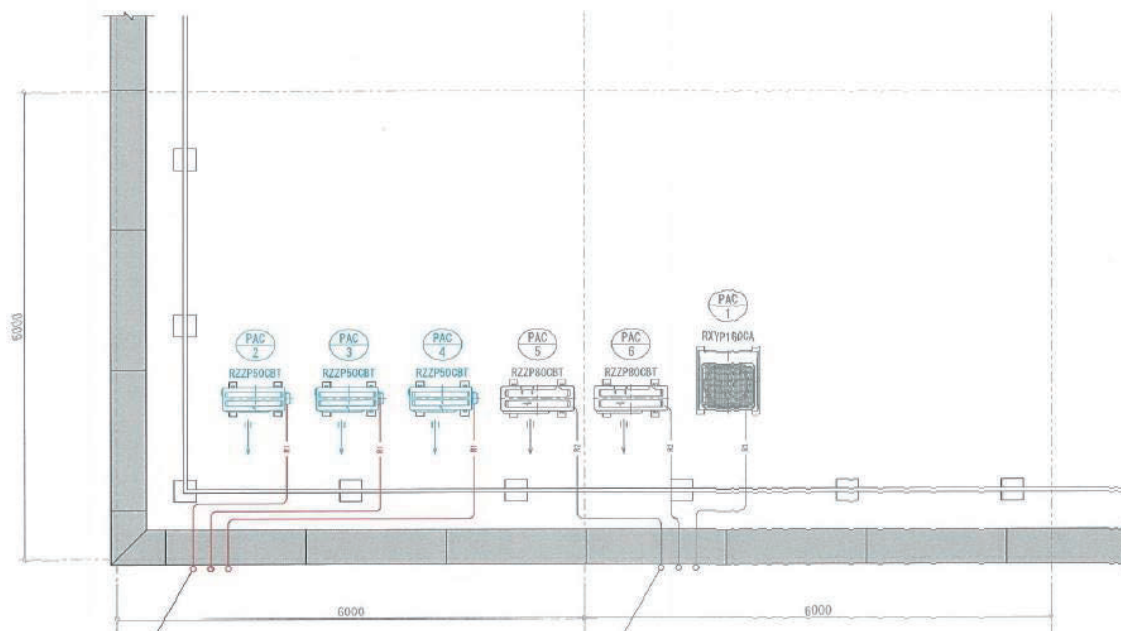


図 69 屋上平面図

(1) 空調システム

室内機の配置を記す。室の運用が模擬オフィスとなっており、様々な試験に対応する為の空調機施工を実施した。窓面の下から上へ送風するペリメーター空調、窓面の上から下面へ送風するペリメーター空調、部屋中心部を空調するインテリア空調、換気装置として、全熱交換器（気化式加湿器内蔵）と、ヒートポンプデシカント調湿外調機をそれぞれ据え付けた。表に室外機と室内機の機器仕様を記す。

表 28 熱源（室外機）仕様

室外ユニット：PAC1	
室外機(形式)：マルチパッケージ形空気調和機	
機器名称	ビル用マルチ Ve-upIV
型番	RXYP160ACA
機器仕様	定格冷暖房能力(冷房/暖房)：16.0 / 18.0kW 定格消費電力(冷房/暖房)：4.31 / 4.02kW 室外ユニット COP：(冷房/暖房)3.71 / 4.48 定格試験条件：JIS B 8615 冷房_室内 27°CDB19°CWB、外気 35°CDB JIS B 8615 暖房_室内 20°CDB、外気 7°CDB,6°CWB ※各室内ユニットまでの配管相当長 7.5m、高低差 0m、ラウンドフロー室内ユニット 100%接続時の値 圧縮機電動機出力：2.9kW,ファン電動機出力：0.30kW 電源：三相 200V 冷媒配管：液側φ9.5、吸入ガス側φ19.1

表 29 室内機仕様

室内ユニット：PAC1-A,B,C,D,D,F	
機器名称	天井埋込ダクト形
型番	FXYLP28MA
機器仕様	定格冷暖房能力：2.8 / 3.2kW 定格消費電力(冷房/暖房)：0.039 / 0.039kW 定格試験条件：JIS B 8616 冷房：室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房：室内 20°CDB、外気 7°CDB,6°CWB 定格機外静圧：100Pa ファン電動機出力：0.015kW 電源：単相 200V 冷媒配管：液側φ 6.4、ガス側φ 12.7mm ファン風量(強-弱)：7-6m ³ /min
室内ユニット：PAC1-G	
機器名称	外気処理機(全熱交換器加湿内蔵タイプ)
型番	VKMP50HM
機器仕様	外気負荷冷暖房処理能力(全熱交換換気-普通換気) : 4.83(2.03)/ 5.76(2.56)kW 定格試験条件：JIS B 8616 冷房_室内 27°CDB,19°CWB、外気 35°CDB JIS B 8616 暖房_室内 20°CDB、外気 7°CDB,6°CWB 加湿方式：流下式自然蒸発加湿器 加湿量：2.9<4.2>kg/h ※加湿量<>内の数値は全熱交による湿度回収分を含んだ値 温度交換効率(特強-強-弱)：76-76-77% 冷房時エンタルピー交換効率(特強-強-弱)：68-68-71% 暖房時エンタルピー交換効率(特強-強-弱)：72-72-74% ファン電動機出力：0.280×2kW 電源：単相 200V 冷媒配管：液側φ 6.4、ガス側φ 12.7mm ファン風量(特強-強-弱)：500-500-400m ³ /h 機外静圧(特強-強-弱)：225-170-125Pa

表 30 熱源（室外機）仕様

室外ユニット：PAC5、6	
機器名称	天井埋込ダクト形(ペア)
型番	RZZP80CBT
機器仕様	定格冷暖房能力(冷房/暖房)：7.1 / 8.0kW 定格消費電力(冷房/暖房)：2.09 / 2.25kW 室外ユニット COP：(冷房/暖房)3.40 / 3.56 定格試験条件：JIS B 8615 冷房_室内 27°CDB19°CWB、外気 35°CDB JIS B 8615 暖房_室内 20°CDB、外気 7°CDB,6°CWB ※各室内ユニットまでの配管相当長 7.5m、高低差 0m、JIS B 8615-1 に準拠 圧縮機電動機出力：1.62kW,ファン電動機出力：0.094kW 電源：三相 200V 冷媒配管：液側 φ 9.5、吸入ガス側 φ 15.9

表 31 室内機仕様

室内ユニット：PAC5、6	
機器名称	天井埋込ダクト形
型番	FHMP80
機器仕様	ファン電動機出力：0.35kW 電源：三相 200V 冷媒配管：液側 φ 9.5、ガス側 φ 15.9mm ファン風量(急-強-弱)：18-16.5-15m ³ /min

表 32 熱源（室外機）仕様

室外ユニット：PAC2、3、4	
機器名称	天井埋込ダクト形(ペア)
型番	RZRP50BT
機器仕様	定格冷暖房能力(冷房/暖房)：4.5 / 5.0kW 定格消費電力(冷房/暖房)：1.58 / 1.25kW 室外ユニット COP：(冷房/暖房)2.85 / 4.00 定格試験条件：JIS B 8615 冷房_室内 27°CDB19°CWB、外気 35°CDB JIS B 8615 暖房_室内 20°CDB、外気 7°CDB,6°CWB ※各室内ユニットまでの配管相当長 7.5m、高低差 0m、JIS B 8615-1 に準拠 圧縮機電動機出力：0.85kW,ファン電動機出力：0.064kW 電源：三相 200V 冷媒配管：液側 φ 6.4、吸入ガス側 φ 12.7

表 33 室内機仕様

室内ユニット：PAC2、3、4	
機器名称	天井埋込ダクト形
型番	FHMP50DB
機器仕様	ファン電動機出力：0.14kW 電源：三相 200V 冷媒配管：液側φ6.4、ガス側φ12.7mm ファン風量(急-強-弱)：16-13-11m ³ /min

表 34 水配管レス調湿外気処理機

水配管レス調湿外気処理器	
機器名称	ヒートポンプデシカント調湿外調機
型式	HDMP50C
機器仕様	定格除湿冷房能力：5.5kW(全熱)、除湿量 6.67kg/h 定格加湿暖房能力：7.1kW(全熱)、加湿量 3.78kg/h 定格試験条件：冷房除湿 室内 27°CDB,19°CWB、外気 33°CDB,28°CWB 暖房加湿 室内 22°CDB,15.5°CWB、外気 0°CDB,-3°CWB 定格消費電力(冷房除湿/暖房加湿)：1.1 / 1.54kW 定格風量：500m ³ /h ファン電動機出力：140×2W 電源：単相 200V

3.1.4 計測概要

3.1.4.1 東京Aビル

(1) 設置状況

3.1.3 項で述べたように、東京Aビルには、室内機×3台、外調機×2台に計測器を設置し能力を測定した。図70に天井裏設置、図71に計測点を番号で記す。



図70 センサー設置位置（天井裏）

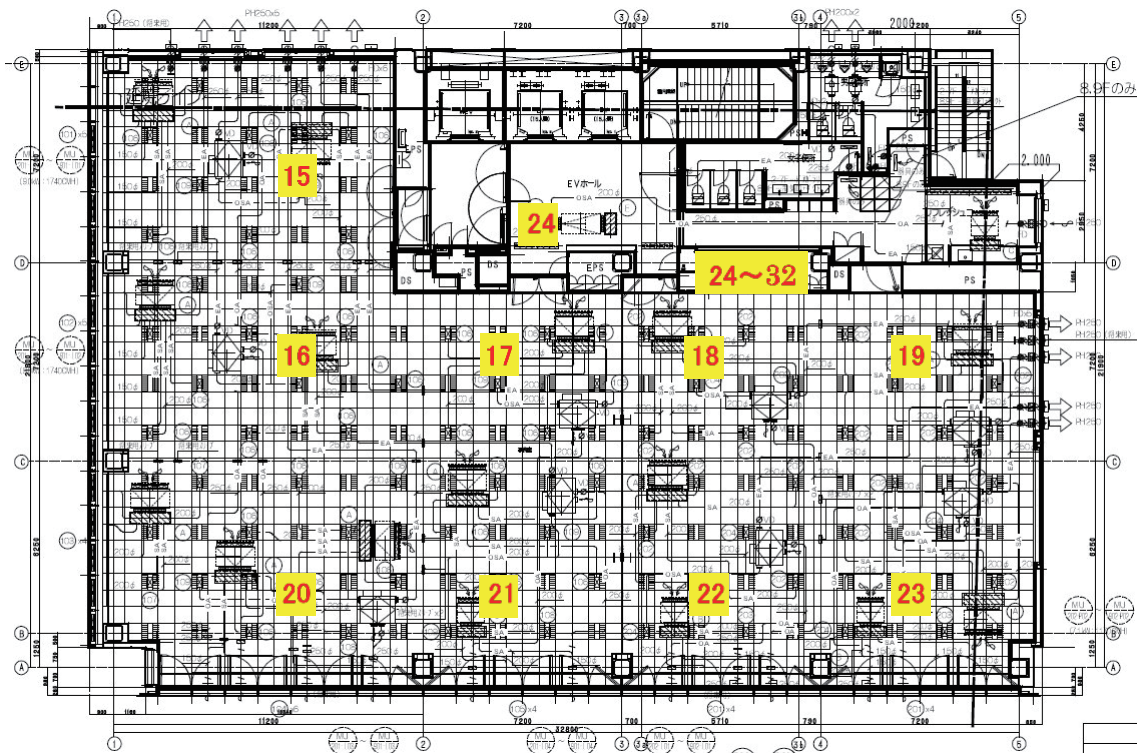


図 71 センサー設置位置

各計測点について詳細内容を以下の表 35 に示す。

表 35 計測点詳細

番号	場所	計測装置	目的
1	空調機 (P01) 吸込口	温湿度センサー (TESTO testo6651/ testo6600、 testo6681/ testo6614) ⇒※マークが testo6681/ testo6614 を使用	室内機、換気装置能力測定 (空気エンタルピー法)
2	空調機 (P01) 吹出口※		
3	換気装置 (D01) 排気口		
4	換気装置 (D01) 給気口		
5	換気装置 (D01) 外気口		
6	換気装置 (D01) 還気口		
7	空調機 (P05) 吸込口		
8	空調機 (P05) 吹出口※		
9	空調機 (I04) 吸込口		
10	空調機 (I04) 吹出口※		
11	換気装置 (D04) 給気口		
12	換気装置 (D04) 還気口		
13	換気装置 (D04) 排気口		
14	換気装置 (D04) 外気口		

15	室内温湿度 1	TR-72wf-H	室内温湿度測定
16	室内温湿度 2		
17	室内温湿度 3		
18	室内温湿度 4		
19	室内温湿度 5		
20	室内温湿度 6		
21	室内温湿度 7		
22	室内温湿度 8		
23	室内温湿度 9		
24	EV ホール温湿度		
25	T1 側空調機-1	消費電力量	空調機器消費電力測定
26	T1 側空調機-2		
27	T2 側空調機		
28	換気装置-1		
29	換気装置-2		
30	換気装置-3		
31	換気装置-4		
32	換気装置-5		
33	換気装置-6		
34	換気装置-7		
35	室内空調機 (P01)		
36	室内空調機 (P05)		
37	室内空調機 (I04)		
38	室外機		
39	室内空調機 (P01)	RAM モニタ	運転データ詳細
40	室内空調機 (P05)		
41	室内空調機 (I04)		
42	室外機		
サンプリングレート		1~14	6 秒
		15~24	600 秒
		25~38	60 秒
		39~41	30 秒
		42	20 秒

写真 1～写真 4 に天井裏に設置した各センサーを記す。各室内機、外調機の空気吸込み側と空気吹出し側にそれぞれセンサーを設置した。



写真 1 室内機吸込み口センサー設置箇所



写真 2 室外機ダクト吹出し口センサー設置箇所



写真 3 外調機吸込み口センサー設置箇所



写真 4 外調機吹出し口センサー設置箇所

室内機入口出口にセンサーを付けたイメージを図 72 に示す。空気エンタルピー法を用いて潜熱能力、顕熱能力を算出するため、測定を実施した空調機は全て同様の施工を行った。

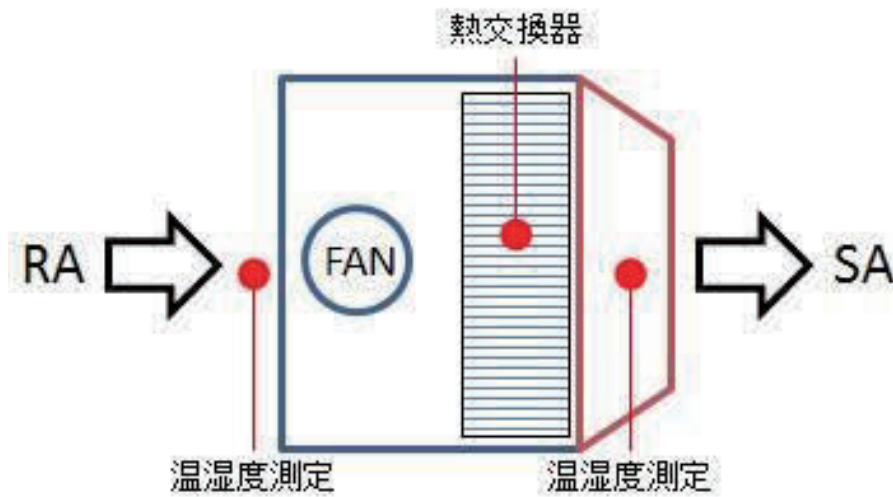


図 72 空調機能力測定

外調機能力測定イメージを、図 73 に示す。

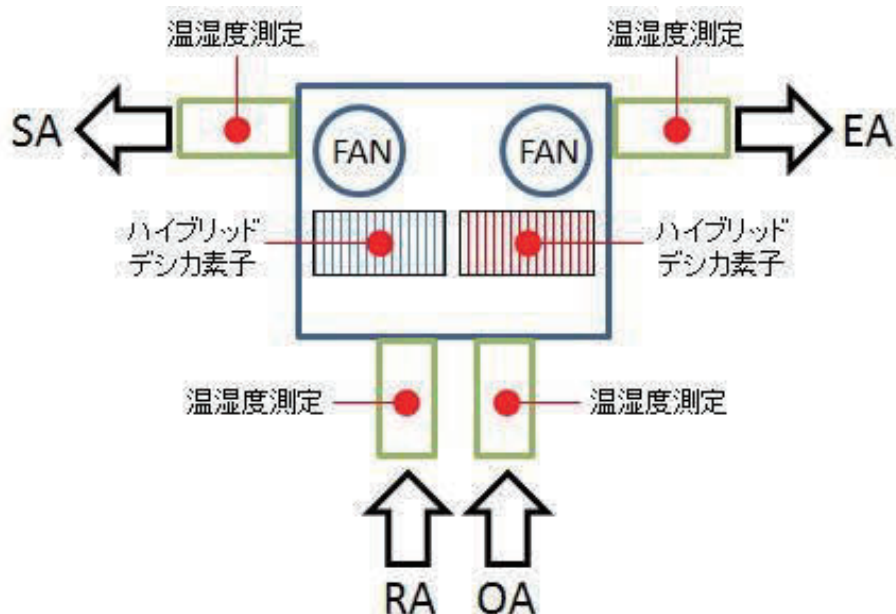


図 73 外調機能力測定

外調機は、室内に除加湿（調湿）を行った空気を供給する。また、還気は外に排気をする。よって、【外気、給気、還気、排気】の値を持ってヒートポンプデシカント調湿外調機の能力を計測する。

尚、執務室に設置した温湿度センサーは執務スペースであり、情報漏えい防止のため写真を撮る事が不可能な為、弊社で同様にセンサーを設置したイメージ写真を写真 6 に記す。室内に 9 か所設置し、エレベーターホールに 1 か所設置した。

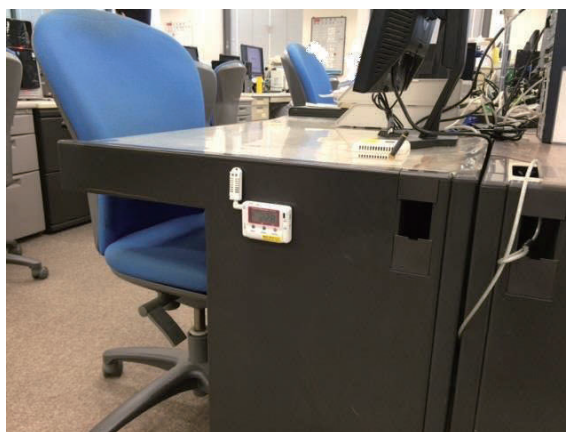


写真 5 机脇に設置イメージ



写真 6 計測器設置イメージ

(2) 計測器精度

表 36 に設置したセンサーの精度を記す。

表 36 センサー精度

メーカー名	型番	仕様
TESTO	testo6651/testo6600	温度測定範囲：-20～70℃ 精度：±0.2℃ (25℃) 湿度測定範囲：0～100%RH 精度：±1.7%RH (0～90%RH) ±1.9%RH (90～100%RH)
	testo6681/testo6614	温度測定範囲：-40～180℃ 精度：±0.15℃ (25℃) 湿度測定範囲：0～100%RH 精度：±(1.0%RH+0.007×計測値)
T and D	TR-72wf-H	温度測定範囲：-30～80℃ 精度：±0.3℃ (0～50℃) ±0.5℃ (その他) 湿度測定範囲：0～99%RH 精度：±2.5%RH (25℃、10～85%RH) ±4.0%RH (25℃、0～10%RH、 85～99%RH) ※25℃以外は上記制度に±0.1%RH/℃を加算 湿度ヒステリシス±1.5%RH
Omron	KM1-PMU2A-FLK (マスタ) KE1-CTD8E (スレーブ) KM20-CTF-50A (CT)	電圧確度：±1.0%FS±1digit 電流確度：±1.0%FS±1digit 有効電力確度：±2.0%FS±1digit 無効電力確度：±2.0%FS±1digit

(3) データ計測システム

データ計測システム概要を以下に記す。本研究では、実験データを遠隔でモニタリングした。図 74 にシステム概略図を記す。

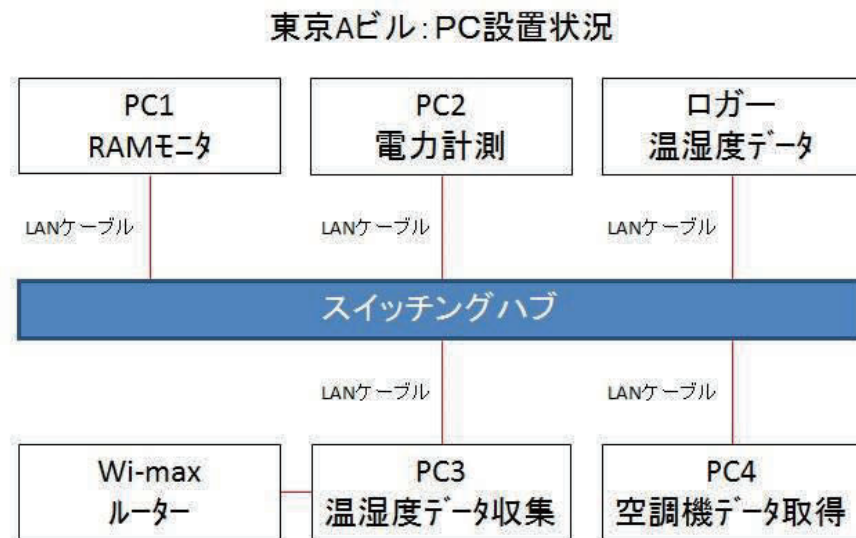


図 74 データ取得システム

3.1.4.2 東京 B ビル

(1) 設置状況

東京 B ビルには室内機×2 台、換気装置×1 台に計測器を設置し能力を測定した。図 75 センサー設置位置（天井裏）に天井裏設置、図 76 センサー設置位置（執務室、EPS）に執務室設置の計測点を番号で記す。

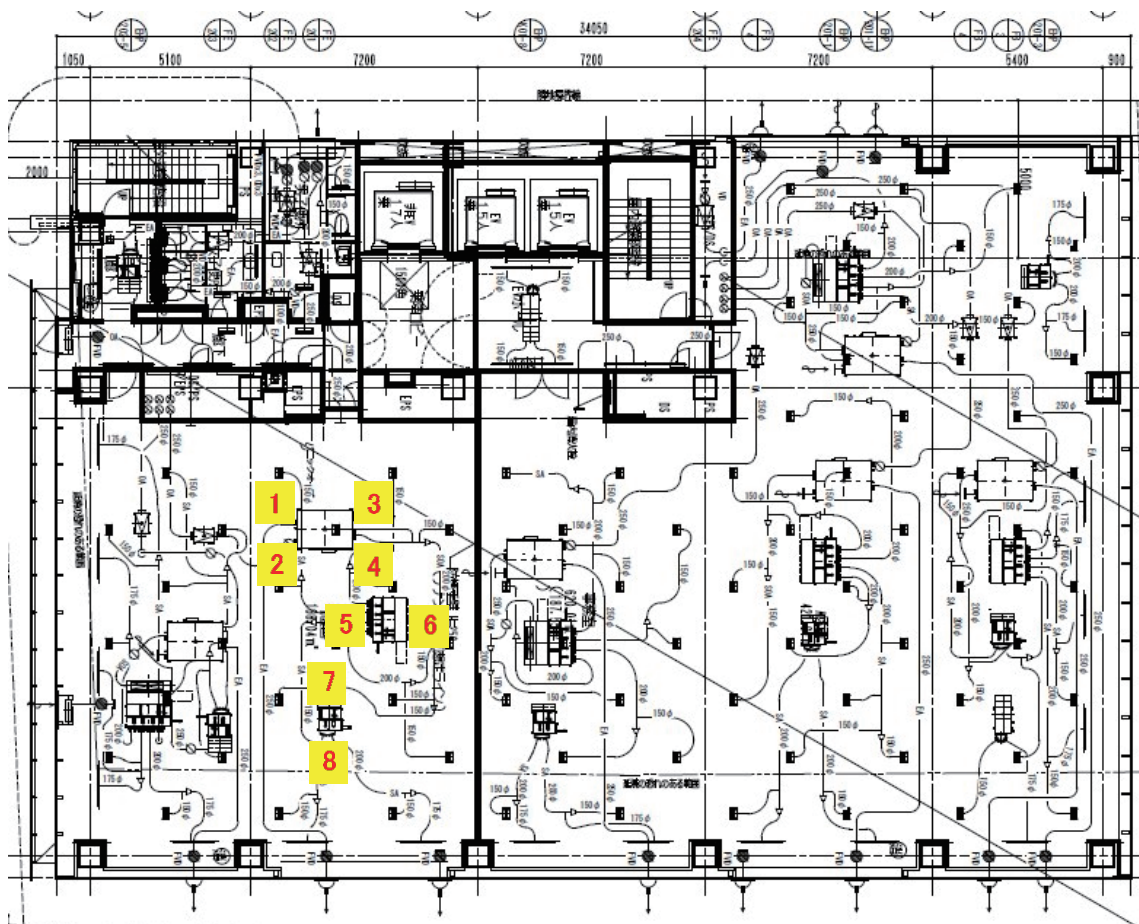


図 75 センサー設置位置（天井裏）

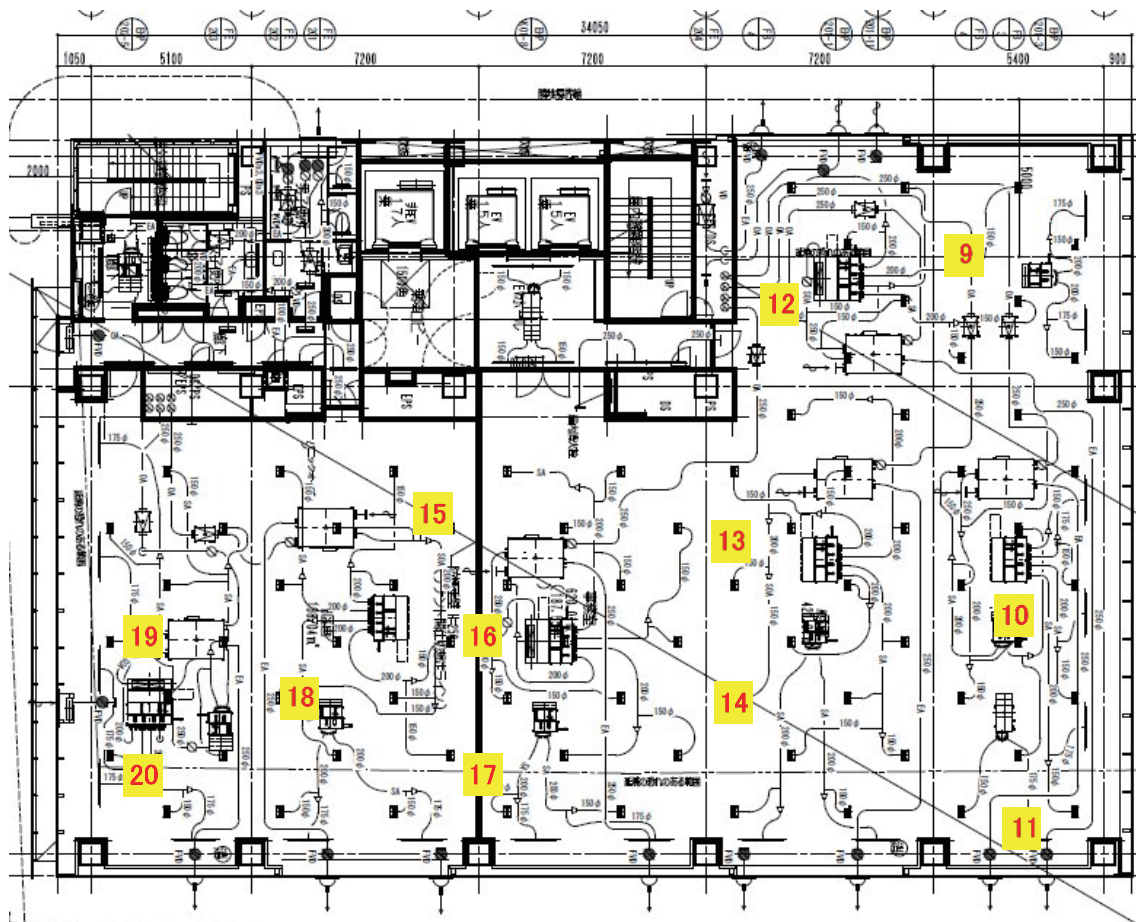


図 76 センサー設置位置（執務室、EPS）

各計測点についての詳細内容を表 37 に示す。

表 37 計測点詳細

番号	場所	計測装置	目的
1	換気装置 (D01) 排気口	温湿度センサー (TESTO testo6651/ testo6600、 testo6681/ testo6614) ⇒※マークが testo6681/ testo6614 を使用	室内機、換気装置 能力測定 (空気エンタルピー 法)
2	換気装置 (D01) 外気口		
3	換気装置 (D01) 還気口		
4	換気装置 (D01) 給気口		
5	空調機 (I01) 吹出口※		
6	空調機 (I01) 吸込口		
7	空調機 (P01) 吹出口※		
8	空調機 (P01) 吸込口		
9	室内温湿度 1	TR-72wf-H	室内温湿度測定
10	室内温湿度 2		
11	室内温湿度 3		
12	室内温湿度 4		
13	室内温湿度 5		
14	室内温湿度 6		
15	室内温湿度 7		
16	室内温湿度 8		
17	室内温湿度 9		
18	室内温湿度 10		
19	室内温湿度 11		
20	室内温湿度 12		
サンプリングレート		1～8	6 秒
		9～20	600 秒

写真 7、写真 8 に天井裏に設置した各センサーの設置状態を示す。各室内機、換気装置の空気吸込み側と空気吹出し側にそれぞれセンサーを設置した。



写真 7 室内機吸込み口センサー設置位置



写真 8 室内機吹出しセンサー設置位置

写真 9、写真 10 に外調機に設置した各センサー設置状態を示す。



写真 9 吸込み口センサー設置位置

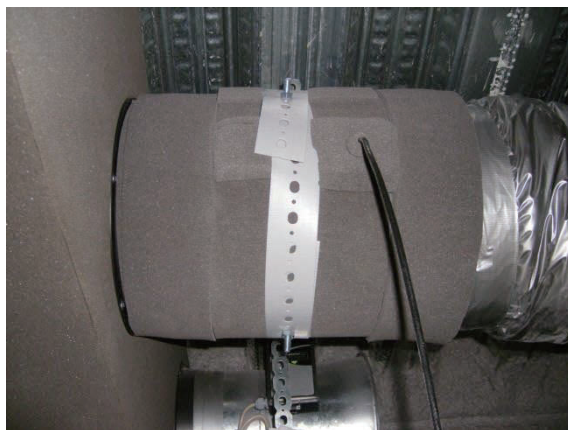


写真 10 外調機吹出しセンサー設置位置

外調機の能力測定イメージを図 77 に示す。

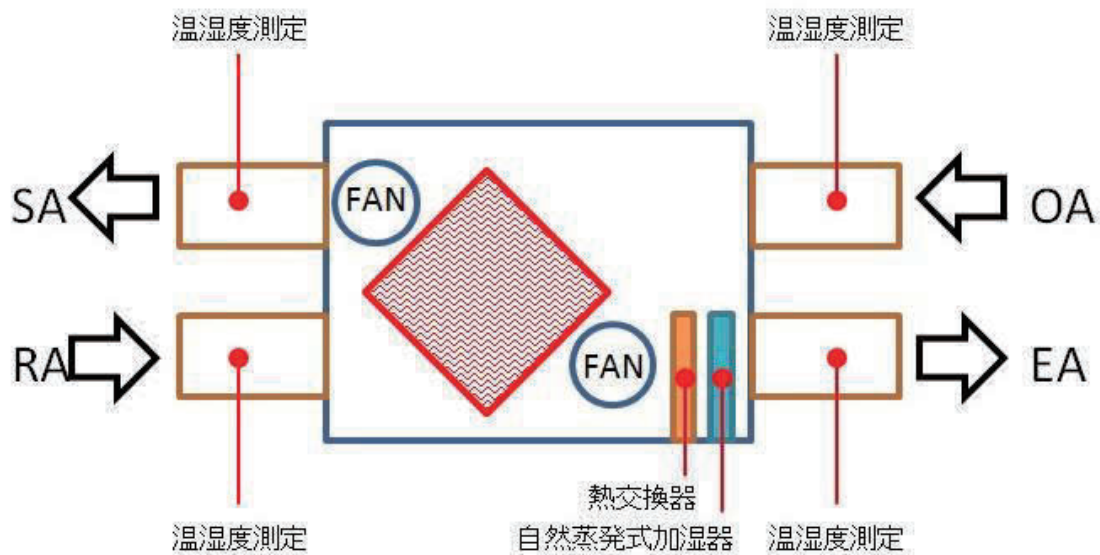


図 77 外調機測定点

VKMP100GAM は熱交換器と、自然蒸発式加湿器が機内に設置されており、全熱回収エレメントの効率を測定するためには、暖房、加湿、冷却、除湿を実施した後の値が給気口の値になる。従って、【還気、排気、外気】の値を持って全熱回収エレメント性能を計測する。

尚、執務室に設置した温湿度センサーは執務スペースであり、情報漏えい防止のため写真を撮る事が不可能な為、弊社で同様にセンサーを設置したイメージ写真を写真 11 に記す。

センサーは室内に 12 か所設置した。

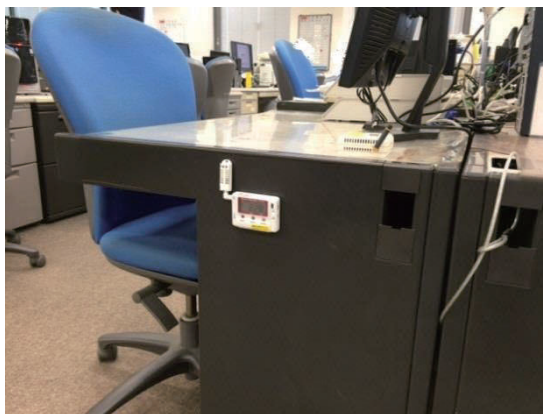


写真 11 執務室センサー取付イメージ

(2) 計測器精度

表 38 にセンサー精度を記す。

表 38 センサー精度

メーカー名	型番	仕様
TESTO	testo6651/testo6600	温度測定範囲：-20～70℃ 精度：±0.2℃ (25℃) 湿度測定範囲：0～100%RH 精度：±1.7%RH (0～90%RH) ±1.9%RH (90～100%RH)
	testo6681/testo6614	温度測定範囲：-40～180℃ 精度：±0.15℃ (25℃) 湿度測定範囲：0～100%RH 精度：±(1.0%RH+0.007×計測値)
T and D	TR-72wf-H	温度測定範囲：-30～80℃ 精度：±0.3℃ (0～50℃) ±0.5℃ (その他) 湿度測定範囲：0～99%RH 精度：±2.5%RH (25℃、10～85%RH) ±4.0%RH (25℃、0～10%RH、 85～99%RH) ※25℃以外は上記制度に±0.1%RH/℃を加算湿度ヒステリシス±1.5%RH

(3) データ計測システム

データ計測システム概要を以下に記す。本研究では、実験データを遠隔でモニタリングした。図 78 にシステム概略図を記す。

東京ビル: PC設置状況



図 78 データ取得システム概要

3.1.4.3 大阪Cビル

(1) 設置状況

大阪Cビルには換気装置×3台に計測器を設置し能力を測定した。

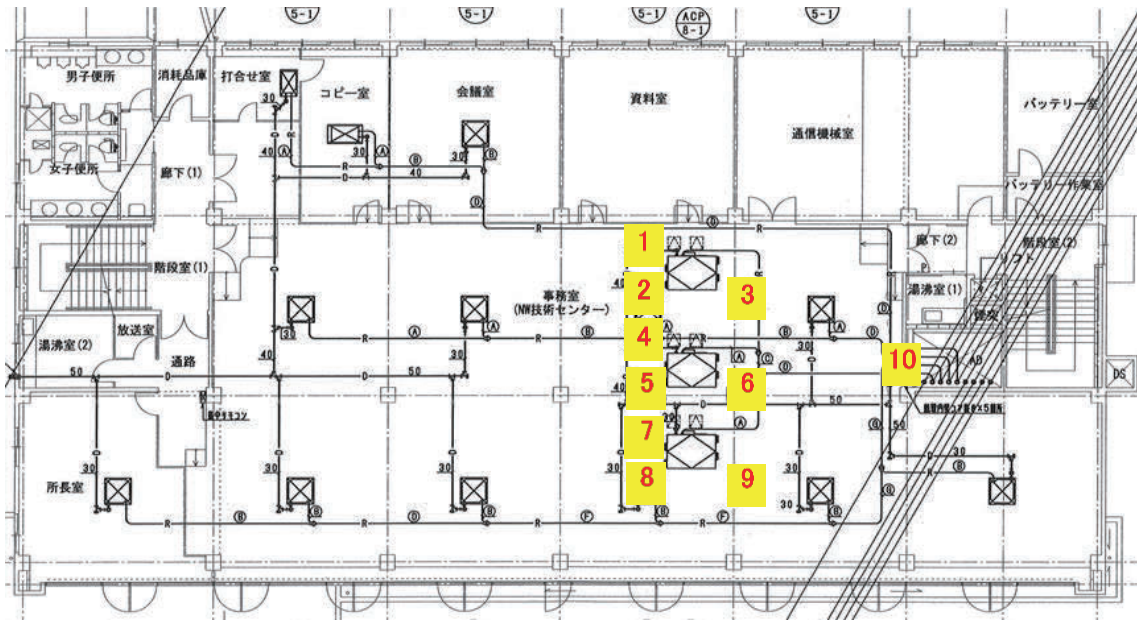


図 79 に天井裏設置、

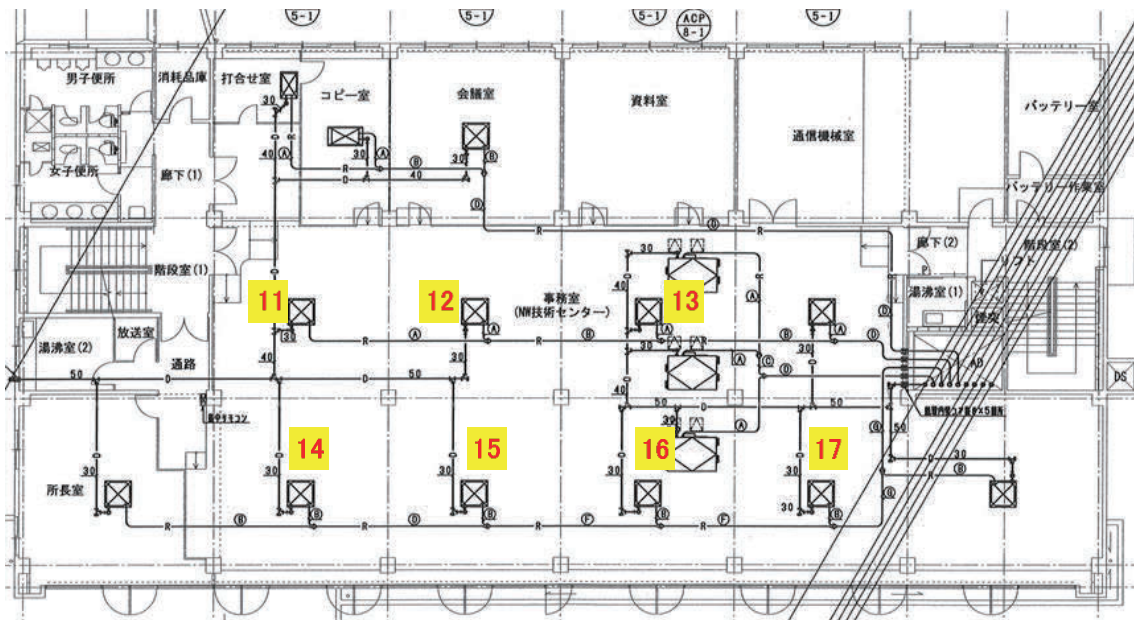


図 80 に執務室設置の計測点を番号で記す。大阪Cビルは外気を縦シャフトでまとめて導入している。その為、外気測定点を一点とした。

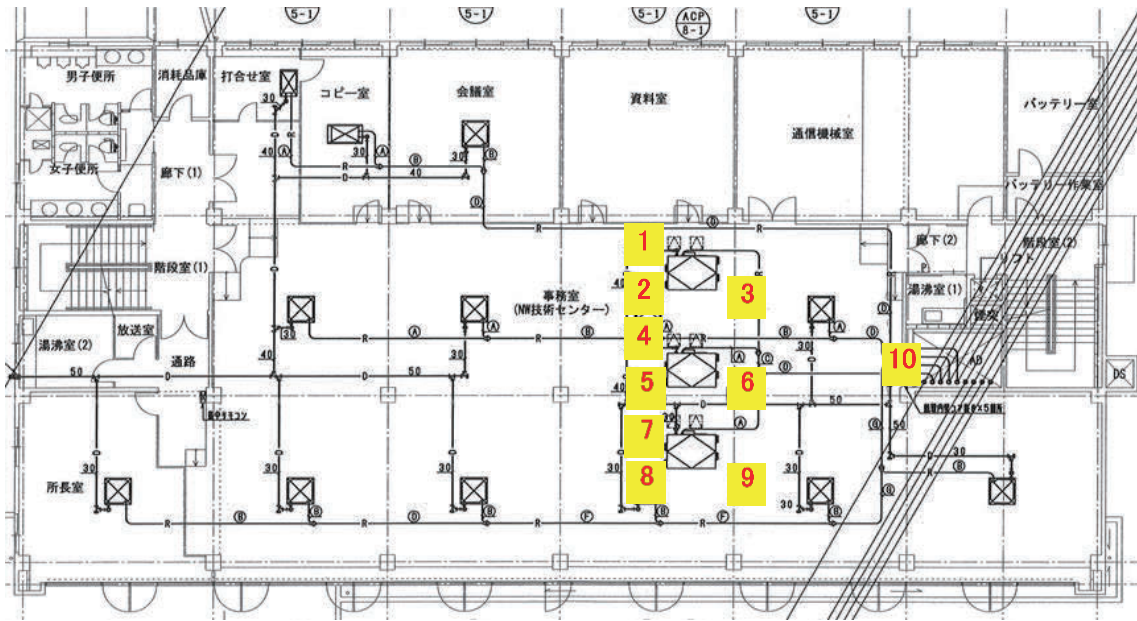


図 79 センサー位置 (天井裏)

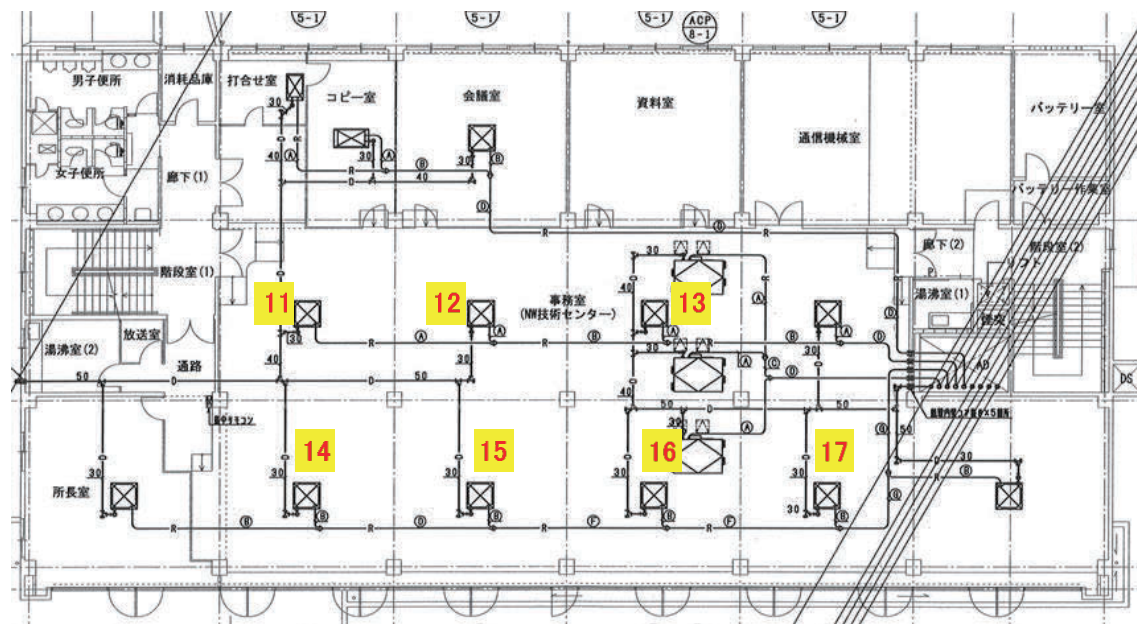


図 80 センサー設置位置 (執務室)

各計測点の詳細内容を表 39 に示す。

表 39 計測点詳細

番号	場所	計測装置	目的
1	換気装置 (D01) 給気口	温湿度センサー (TESTO testo6651/ testo6600)	室内機、換気装置 能力測定 (空気エンタルピー法)
2	換気装置 (D01) 還気口		
3	換気装置 (D01) 排気口		
4	換気装置 (D02) 給気口		
5	換気装置 (D02) 還気口		
6	換気装置 (D02) 排気口		
7	換気装置 (D03) 給気口		
8	換気装置 (D03) 還気口		
9	換気装置 (D03) 排気口		
10	換気装置外気ダクト		
11	室内温湿度 1	TR-72wf-H	室内温湿度測定
12	室内温湿度 2		
13	室内温湿度 3		
14	室内温湿度 4		
15	室内温湿度 5		
16	室内温湿度 6		
17	室内温湿度 7		
サンプリングレート		1~10	6 秒
		11~17	600 秒

写真 12 に換気装置 SA 吹出し側センサー設置状況を記す。各室内機、換気装置のダクト接続口にセンサーを設置した。



写真 12 換気装置 SA 吹出し側センサー設置状況

尚、執務室に設置した温湿度センサーは執務スペースであり、情報漏えい防止のため写真を撮る事が不可能な為、弊社で同様にセンサーを設置したイメージ写真を写真 13 に記す。センサーは、室内に 7 か所設置した。

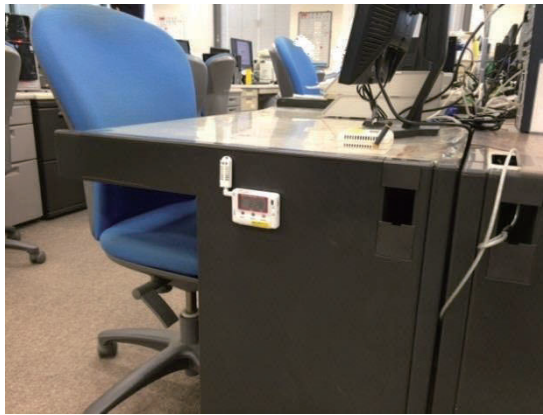


写真 13 執務室センサー取付イメージ

(2) 計測器精度

表 40 にセンサー精度を記す。

表 40 センサー精度

メーカー名	型番	仕様
TESTO	testo6651/testo6600	温度測定範囲：-20～70℃ 精度：±0.2℃ (25℃) 湿度測定範囲：0～100%RH 精度：±1.7%RH (0～90%RH) ±1.9%RH (90～100%RH)
T and D	RTR-53AL	温度測定範囲：0～50℃ 精度：±0.3℃ (25℃) 湿度測定範囲：10～95%RH 精度：±5.0%RH (25℃)

(3) データ計測システム

データ計測システム概要を以下に記す。本研究では、実験データを遠隔でモニタリングした。図 81 にシステム概略図を記す。

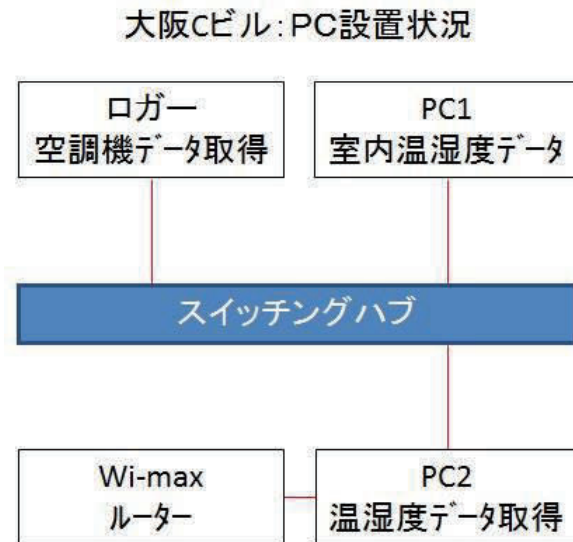


図 81 データ取得システム概要

3.1.4.4 国立研究開発法人 建築研究所 環境実験棟

用いた計測器と据え付けた場所について以下に記す。

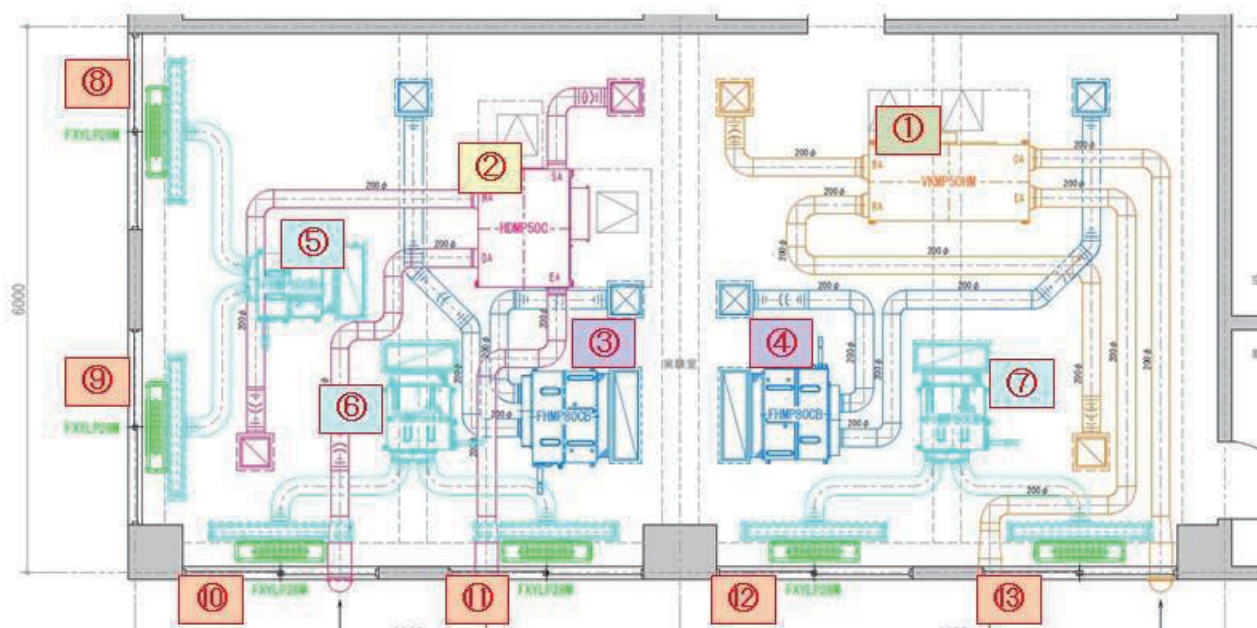


図 82 空調機据付場所 (4 階)

表 41 空調機型番と据付計測器一覧

番号	用途	室内機	データ取得	センサ位置	センサ型番	センサ目的
①	全熱交換器	VKMP50HM	温湿度	OA、SA、RA、EA	TESTO:6651-6600	温湿度交換量、能力算出
			電力	機器本体	omron:KM1-PMU2A-FLK	電力量取得、COP
②	調湿換気装置	HDMP50C	温湿度	OA、SA、RA、EA	TESTO:6651-6600	温湿度交換量、能力算出
			電力	動力盤	omron:KM1-PMU2A-FLK	電力量取得、COP
			RAMモニタ	プリント基板	自社製品	能力算出
③~④	インテリア空調機	FHMP80CB	温湿度	OA、SA、RA、EA	TESTO:6651-6600	温湿度交換量、能力算出
			電力	機器本体	omron:KM1-PMU2A-FLK	電力量取得、COP
			RAMモニタ	プリント基板	自社製品	ファン回転数
⑤~⑦	ペリメーター空調機(天井)	FHMP50CB	温湿度	OA、SA、RA、EA	TESTO:6651-6600	温湿度交換量、能力算出
			電力	機器本体	omron:KM1-PMU2A-FLK	電力量取得、COP
			RAMモニタ	プリント基板	自社製品	ファン回転数
⑧~⑬	ペリメーター空調機(床面)	FXYP28M	温湿度	OA、SA、RA、EA	TESTO:6651-6600	温湿度交換量、能力算出
			電力	機器本体	omron:KM1-PMU2A-FLK	電力量取得、COP
			チェッカー	プリント基板	自社製品	ファン回転数

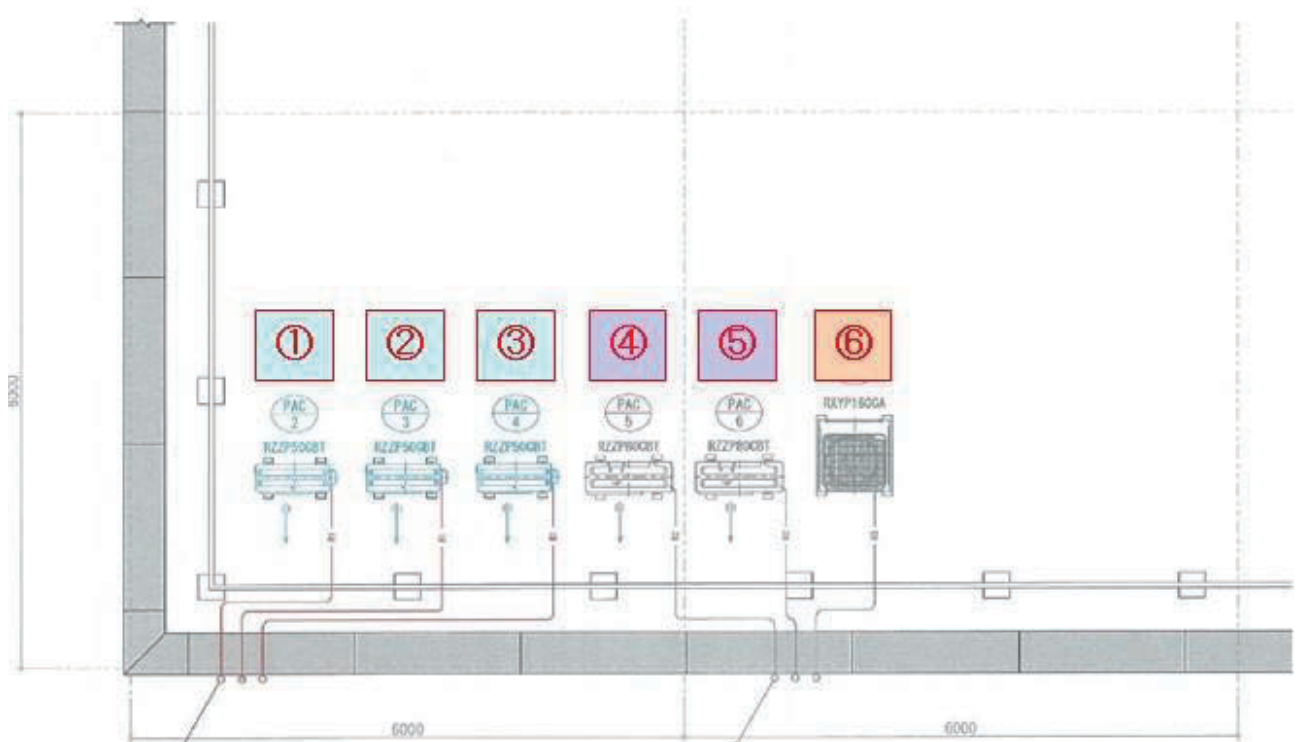


図 83 空調機据付場所（屋上）

表 42 空調機型番と据付計測器一覧

番号	用途	室外機	データ取得	センサ位置	センサ型番	センサ目的
①～③	ペリメーター空調機（天井）	RZP50CBT	配管温度	吐出管、吸入管、アキュム前、電動弁前後、ガス管、液管	T型熱電対	能力算出
			電力	動力盤	omron:KM1-PMU2A-FLK	電力量取得、COP
			RAMモニタ	プリント基板	自社製品	能力算出
④～⑤	インテリア空調機	RZP80CBT	配管温度	吐出管、吸入管、アキュム前、電動弁前後、ガス管、液管	T型熱電対	能力算出
			電力	動力盤	omron:KM1-PMU2A-FLK	電力量取得、COP
			RAMモニタ	プリント基板	自社製品	能力算出
⑥	ペリメーター空調機（床面）	RXP160CA	配管温度	吐出管、吸入管、アキュム前、電動弁前後、ガス管、液管	T型熱電対	能力算出
			電力	動力盤	omron:KM1-PMU2A-FLK	電力量取得、COP
			チェッカー	プリント基板	自社製品	能力算出

図 84 に EA 経路と OA 経路に補助ファン取付位置を記す。また、風量測定器(ニューエアロアイ)は青色枠の部分に取り付けた。また、取り付けたセンサーや補助ファン、CO₂封入位置について下記に記す。これらの補助ファンには全てインバータが接続されている為、任意に風量と静圧を調整することが出来る。

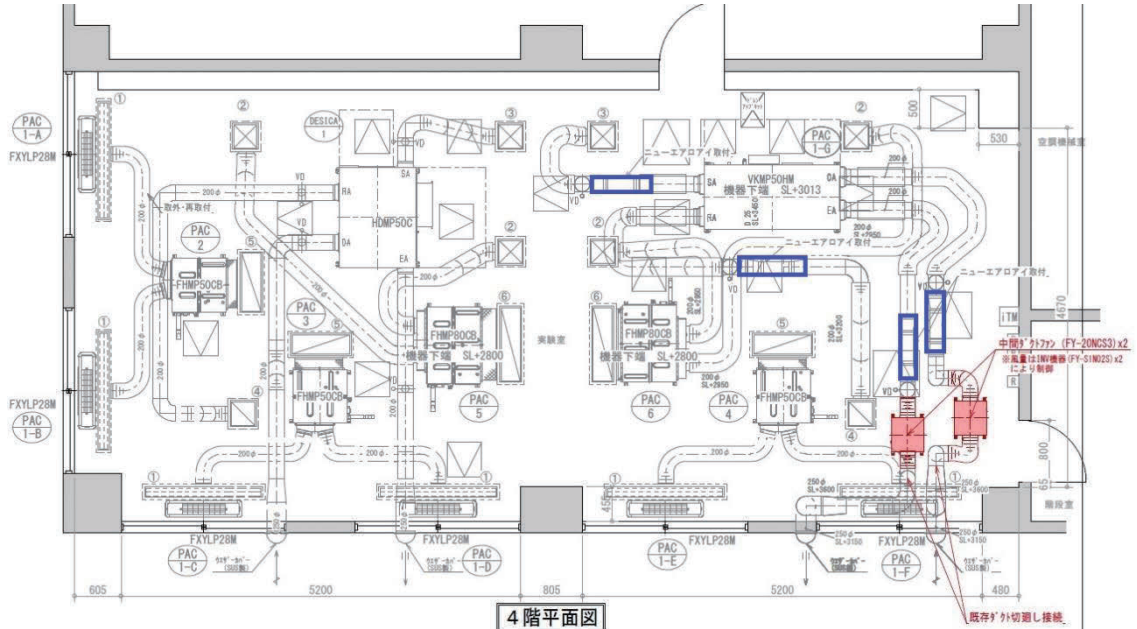


図 84 補助ファン、風量計据付場所 (4 階)

測定器仕様を以下に記す。

表 43 測定機器仕様

メーカー名	型番	仕様
TESTO	testo6651 A01 B01 C00 D01 E00 F01 G02 H00 K06 testo6600 L04 M03 N02 P14	温度測定範囲：-20～70℃ 精度：±0.2℃(25℃) 湿度測定範囲：0～100%RH 精度：±1.7%RH(0～90%RH) ±1.9%RH(90～100%RH)
横河電機	FP201A-E31-LC0A*B/B1	圧力測定範囲：0～5MPa(abs) 精度：±0.25% 温度特性保障範囲：-25～75℃ 動作温度範囲：-40～80℃ 周囲湿度範囲：5～95%RH(結露しない事)
omron	KM1-PMU2A-FLK(マスタ) KE1-CTD8E(スレーブ) KM20-CTF-50A(CT)	電圧確度：±1.0%FS±1digit 電流確度：±1.0%FS±1digit 有効電力確度：±2.0%FS±1digit 無効電力確度：±2.0%FS±1digit
ウエットマ スター	ニューエアロアイ：AE150D	風速：4.0～25.0m/s 風量：254.5～1590.0m ³ /h 温度：0～120℃ 湿度：結露しない事 検出圧力：全圧と動圧 精度：±2%R.D.
長野計器	デジタル微差圧計：GC62	差圧表示精度：±(1.0%F.S.+1digit) at 23℃(0.5kPa 以上) ±(1.5%F.S.+1digit) at 23℃(0.2kPa 以下) 温度係数：±0.1%F.S./℃ ゼロスパン共 使用温度範囲：-10～50℃(凍結・結露無 きこと) 使用湿度範囲：35～85%RH(結露無きこ と)

3.2 エアコンの潜熱処理特性把握および潜熱処理モデルの作成

3.2.1 取組概要

本取組の目的の一つである、潜熱負荷処理メカニズムを考慮した検討を実施するためには、住宅モデルと連携させるための、除湿特性を含む簡易なルームエアコンディショナーモデルを作成する必要がある。簡易なモデル化のためには、特性に影響のある要因の間引き等が必要となるが、そのためにはルームエアコンディショナーの除湿特性を含む詳細モデルを作成し、各要因の除湿特性等に与える影響度合い等を把握する必要がある。また、湿度（潜熱負荷）を考慮した検討を実施する必要があるため、これまでの全熱を対象とした検討に加え、例えば室内機吸込み空気湿度といった新たな解析要素が必要となると考えられる。

しかし、これまでは住宅の冷房負荷として全熱に着目して検討していたため、取得済みの実験データは、室内機の吸込み空気温湿度条件が狭い範囲となっており、湿度要因を検討するために必要と考えている温湿度範囲が不足しており、検討のためにはモデル化に必要な実験データの取得が必要となる。

そこで、モデル案の検証に用いる空調機計測データを取得するための実験装置の整備ならびに実験方法の検討を行い、実験室実験により検証データを取得した。また、一部は解析まで実施した。図 85 に全体イメージを示す。

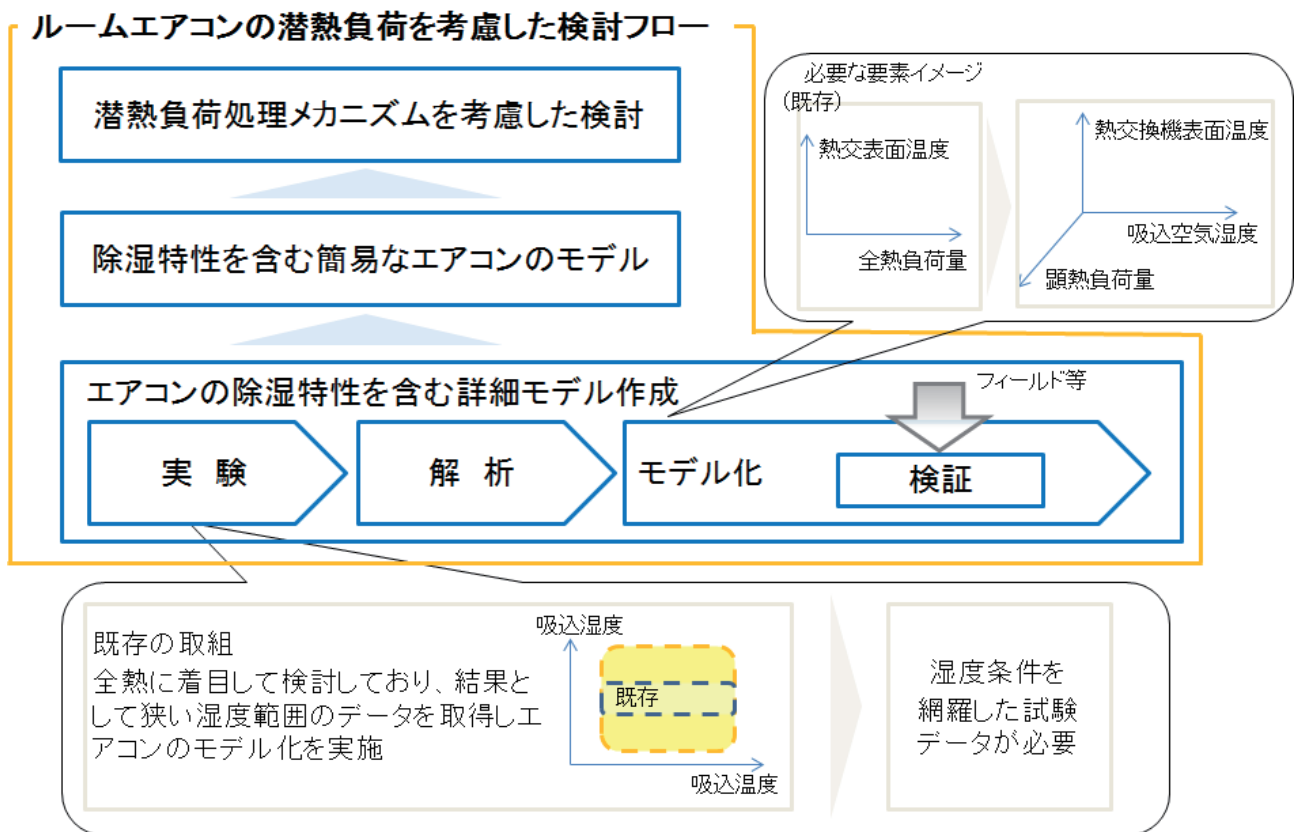


図 85 取組全体イメージ

3.2.2 潜熱処理特性把握実験

3.2.2.1 目的と背景

住宅向け空調機の除湿特性を考慮したモデル作成ならびに検証に使用する、ルームエアコンディショナーの冷房および除湿モードにおける運転データの取得を目的とした。

3.2.2.2 実験概要

湿度を考慮したモデルの検証に利用することを目的としているため、再現性がありできるだけ高い精度のデータとする必要があった。そこで、ルームエアコンディショナーの運転環境を任意に制御できる、人口環境装置を用い、機器の潜顕熱能力は空気エンタルピー法を用いて計測するために室内機をチャンバーに接続して計測した。なお、計測装置及び能力等の計測は JIS B 8615-1 に準拠して実施した。図 86 に実験室の概要を、表 44 に計測点の概要、表 45 に供試機の概要をそれぞれ示す。

本実験では、ルームエアコンディショナーの潜熱負荷処理メカニズムを考慮した特性を把握するために、大きく 3 つの実験を行った。それぞれの特徴を以下に示す。

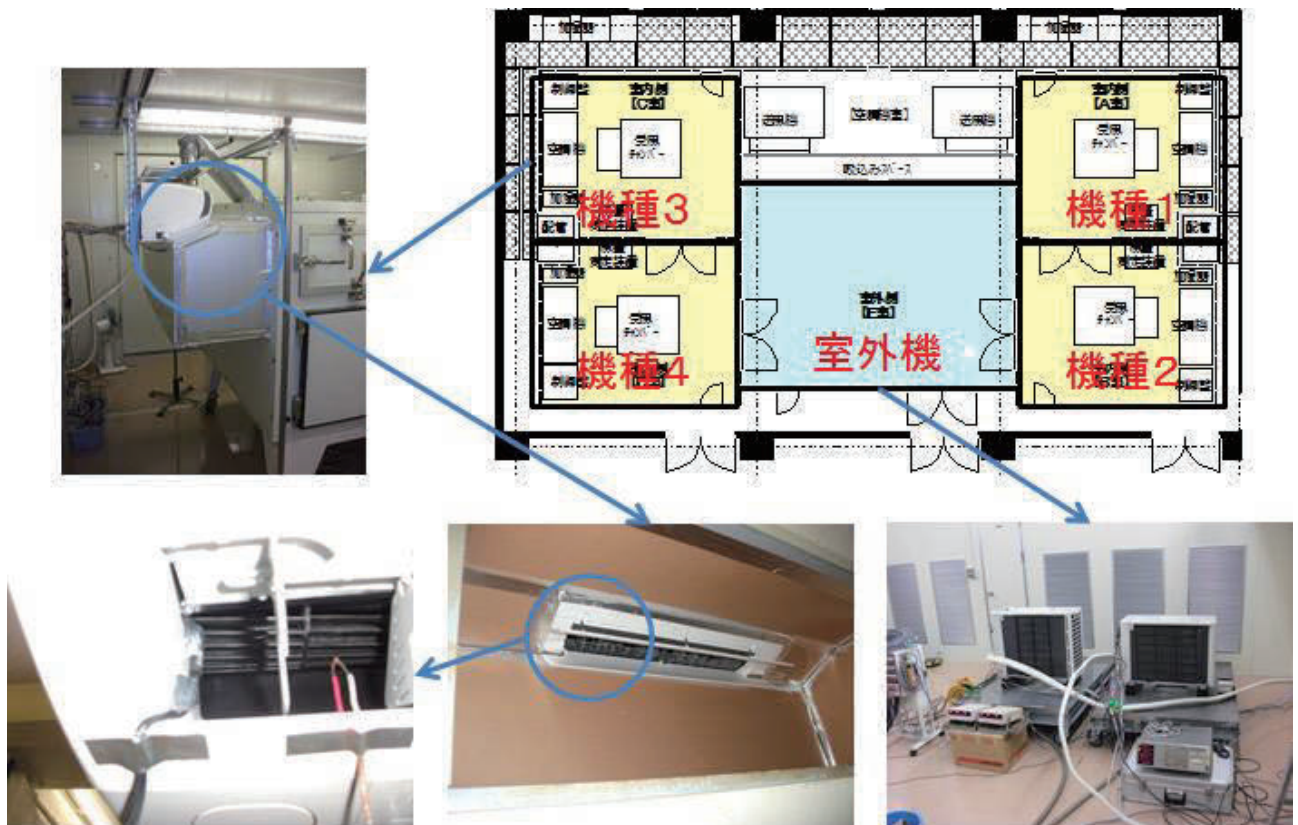


図 86 実験室概要

表 44 計測項目概要

対象	項目	取り付け位置	計測センサー等
空気	温度	室内機吸込	測温抵抗体(Pt100,classA)
		室内機吹出	
		室外機吸込	
		室外機吹出	
	湿度	室内機吸込 (WB)	測温抵抗体(Pt100,classA)
		室内機吹出 (WB 及び相対湿度)	測温抵抗体(Pt100,classA) 高分子薄膜静電容量式センサー (VAISALA HMP155)
		室外機吸込	測温抵抗体(Pt100,classA)
		室外機吹出	
風量	室内機吹出	JISB8615-1 付属書 B による	
エネルギー	電力	全体	電力量計 HIOKI 3390 (圧縮機はクランプ 9272)
		圧縮機	
ファン	送風ファン回転数	室内機	非接触
冷媒	圧力	圧縮機吸込	圧力計(横河電機 FP-101)
		圧縮機吐出	
	温度	圧縮機吸込	T 熱電対
		圧縮機吐出	
		室外機熱交換器入口	
		室外機熱交換器中央	
		室外機熱交換器出口	
		室内機熱交換器入口	T 熱電対
		室内機熱交換器中央	T 熱電対
室内機熱交換器出口	T 熱電対		

表 45 供試機概要

機種番号	冷房定格能力	冷房定格消費電力	メーカー
機種 1	6.3 kW	2450 W	A 社
機種 2	2.2 kW	450 W	A 社
機種 3	6.3 kW	1920 W	B 社
機種 4	2.2 kW	450 W	B 社

3.2.2.3 実験内容

(1) 温湿度変化実験

人工環境装置において室内機設置空間を一定の温湿度に保ち行う実験であり、再現性が高く機器特性を精緻に把握することができる。なお、室外機設置空間の温湿度も人工環境装置で一定に保つ。

除湿特性に影響が大きいと考えられる要因を変化させて実験を行う。変化させるパラメータのイメージを表 46 に、実験に用いた温湿度変化設定の一例を図 87 温湿度変化実験の温湿度設定例に示す。

表 46 運転条件変化概要

パラメータ	パターン		
吸込空気温度	高温	定格条件	低温
吸込空気湿度	高湿	定格条件	低湿
風量	最大	中間	最小
運転モード	冷房	除湿	

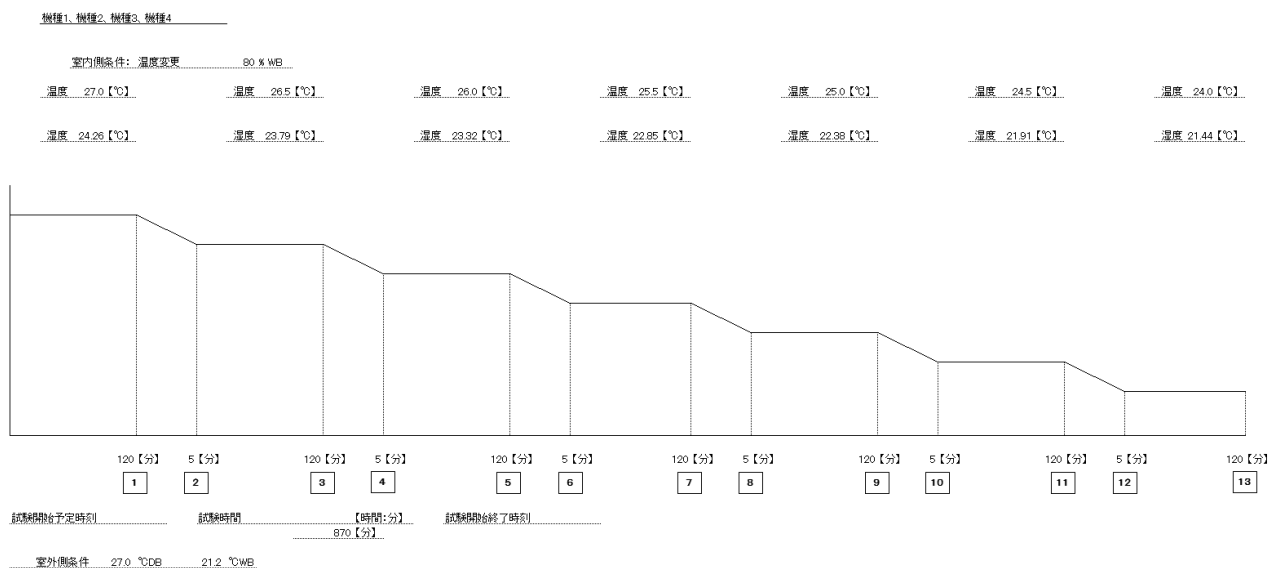


図 87 温湿度変化実験の温湿度設定例

(2) 負荷変化実験

人工環境装置において室内機設置空間へ一定の潜顕熱負荷を与へ、ルームエアコンディショナーにより温湿度をコントロールさせる実験であり、実際の機器挙動に近い運転状況を確認することができる。なお、室外機設置空間の温湿度も人工環境装置で一定に保つ。冷房負荷の大きさは機器の最大能力までを対象とし、潜熱顕熱の割合も機器が処理できる範囲確認と合わせ可能な限り広く変化させた。

実験に用いた負荷設定の一例を図 88 に示す。

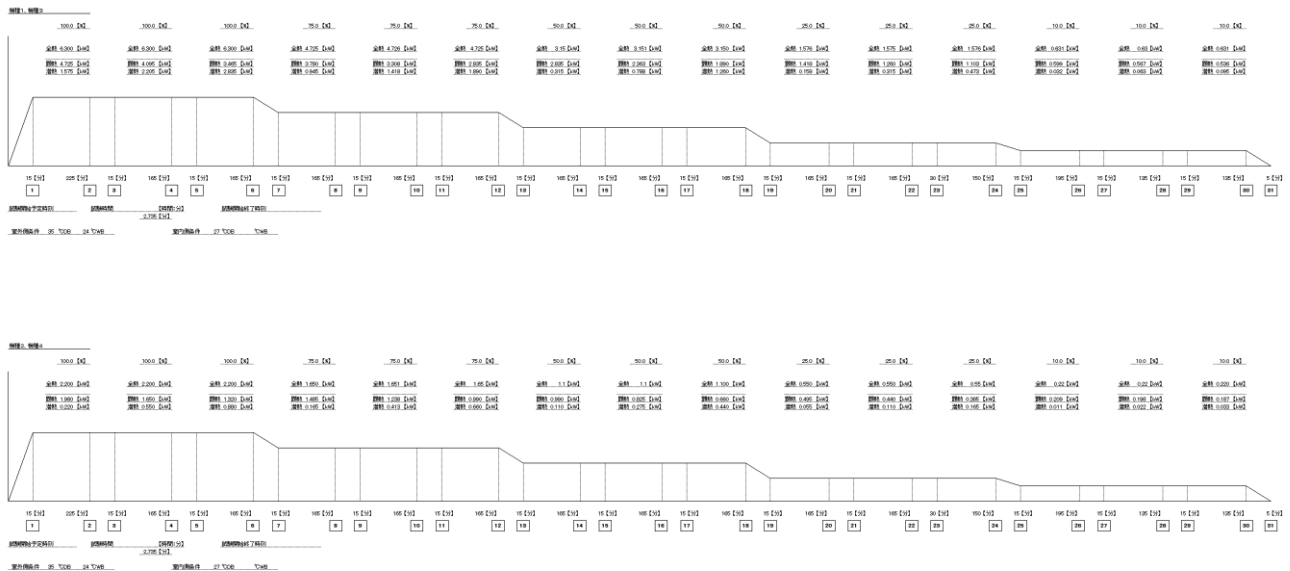


図 88 負荷変化実験の負荷設定例

(3) 室内機内部保水量

実験データの解析を進めたところ、ルームエアコンディショナー室内機内部に除湿によって生じた水分が残留している可能性があったことから2つの実験により確認を行った。

(I) 空気エンタルピー法による実験

室内機設置空間を一定の温湿度に保ち、ルームエアコンディショナーを任意の運転モードで運転させ、保留水の状態を定常とするために凝縮水の排出が十分になされたことを確認した後に停止させ、結露水が自然に排出される時間として20分間そのままの状態に放置する。その後、チャンバーの送風ファンにより吸気し、室内機内部を乾燥させ、空気エンタルピー法により計測した室内機からの放出水分量を積算して内部保留水量を確認する実験。

(II) 重量計測による確認実験

空気エンタルピー法による実験と同様に室内機を準備した後、凝縮水が室内機外へ流れ出さないようにし、ルームエアコンディショナーをチャンバー及び室外機から取外して室内機の重量を測定する。その後、凝縮水が排出されるようにした後に48時間人口環境装置内の湿度を40%に保って放置し内部を乾燥させ、再び室内機の重量を測定して、乾燥の前後における重量差より内部保留水量を確認する実験。本実験はルームエアコンディショナーをチャンバーから取外すだけでなく冷媒回収も行うことから、室内の温湿度条件35℃、47%の1回のみ実施した。状況を写真14に示す。



写真 14 重量測定状況

(4) フィールド模擬実験

実験室で確認された保水量について、新たに確認された内容であったため実験室特有の事象でないことなどを確認するために、フィールド模擬実験を行った。

(I) 実験概要

実験室実験で計測した計測点のうち、チャンバーに付随する計測以外は同様の計測点とし、追加して、室内機と室外機の総重量、ならびに室内機からの排水量を計測することにより、重量変化より室内機内部の保水量を確認する。

(II) 実験装置概要

環境実験室内に設置された、実験住宅の一室を用いて行った。住宅の平面図を以下に示す。図面左側の空間に、エアコン、計測装置、人体を模擬した発湿装置などを設置した。

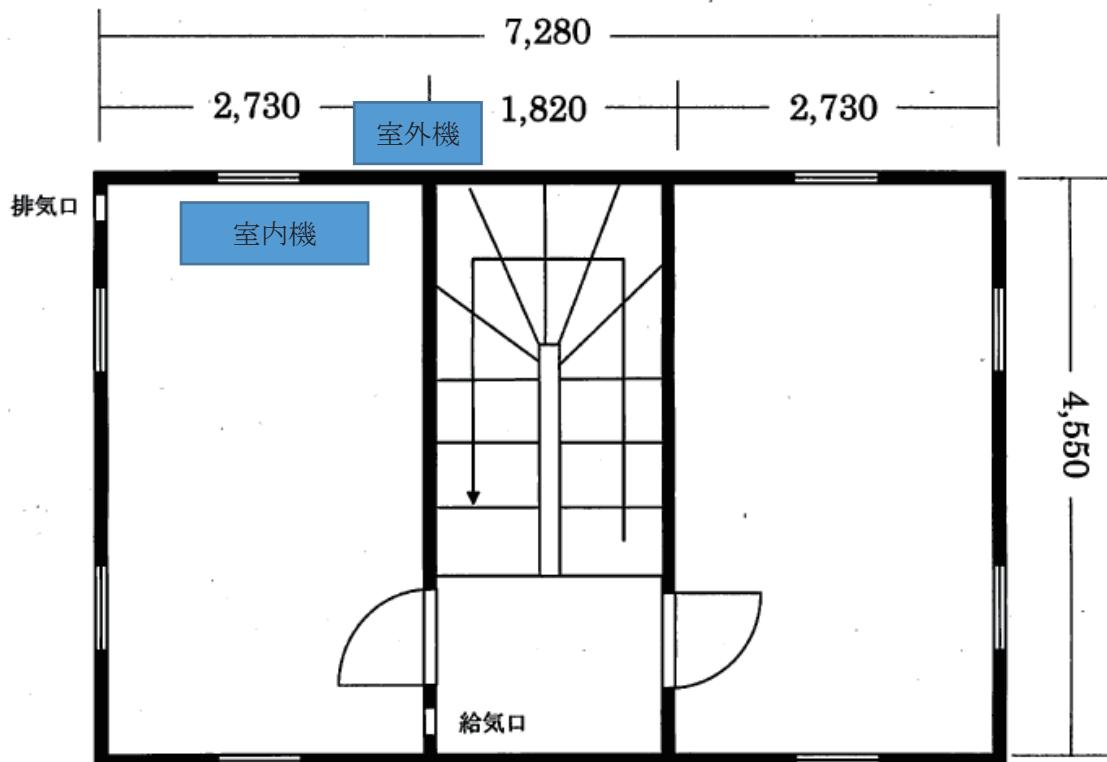


図 89 実験平面図

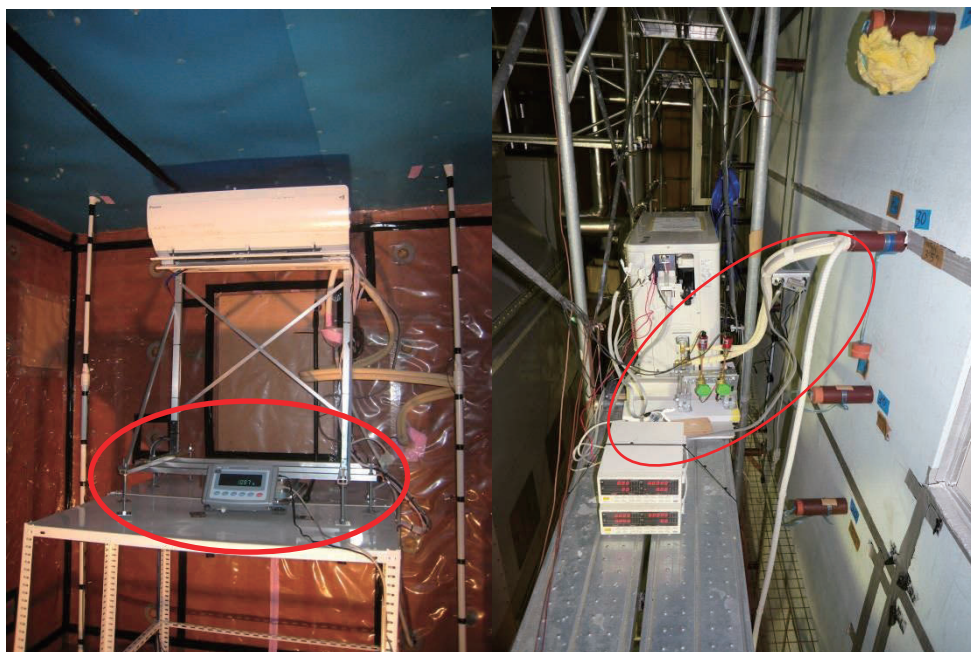


写真 15 エアコンの重量計測状況（左：室内機、右：室外機）

3.2.2.4 実験結果

それぞれの実験方法における結果の一部を示す。

(1) 温湿度変化実験

室内温度を 29 °C 一定とし、湿度を変化させた際の時系列データを図 90 から図 93 に機種ごとに示す。なお、上段にルームエアコンディショナーの吸込吹出温湿度を、下段に潜熱処理量と風量を示す。温度および湿度が変化に合わせ処理熱量が変化していることがわかる。

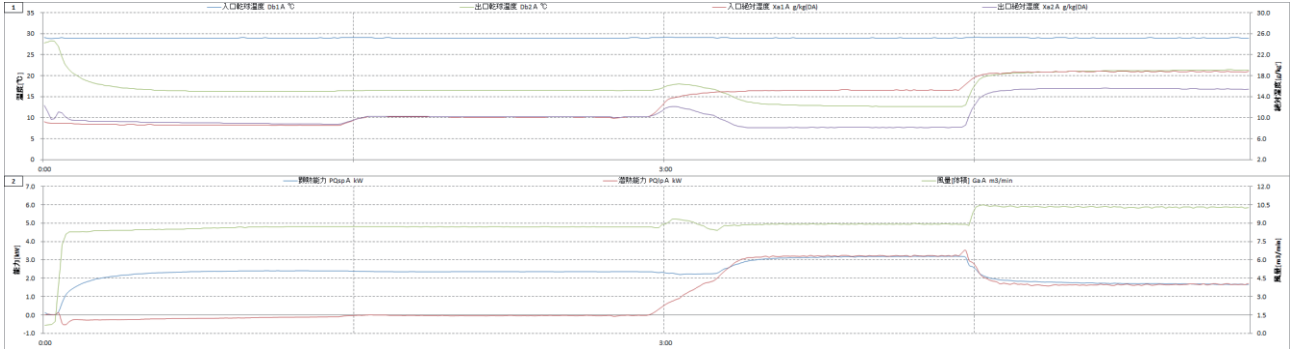


図 90 機種 1 の結果例

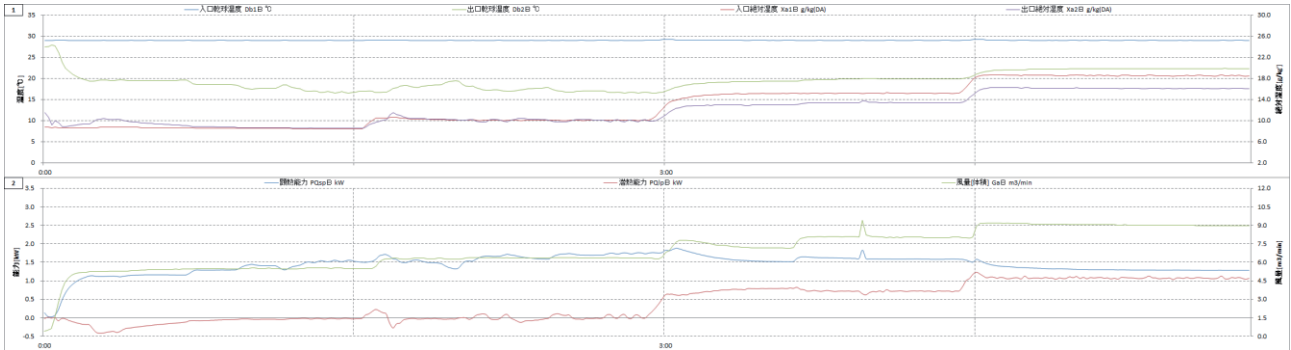


図 91 機種 2 の結果例

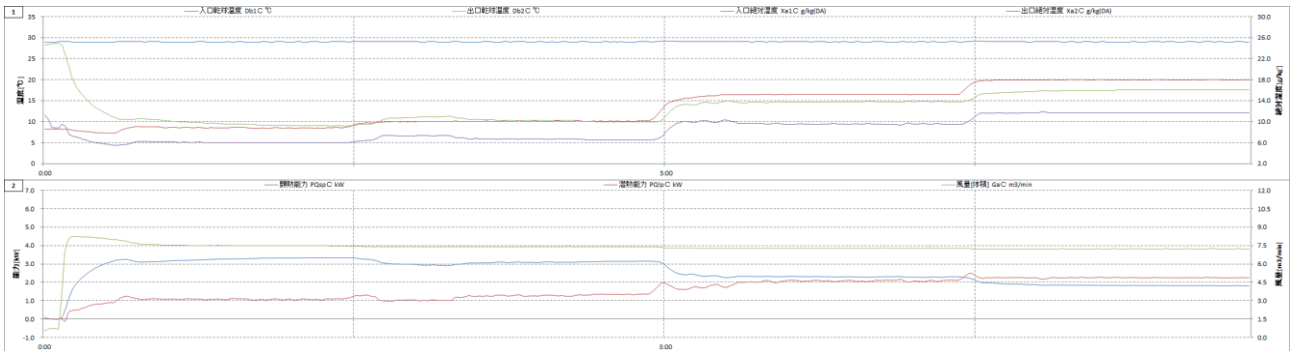


図 92 機種 3 の結果例

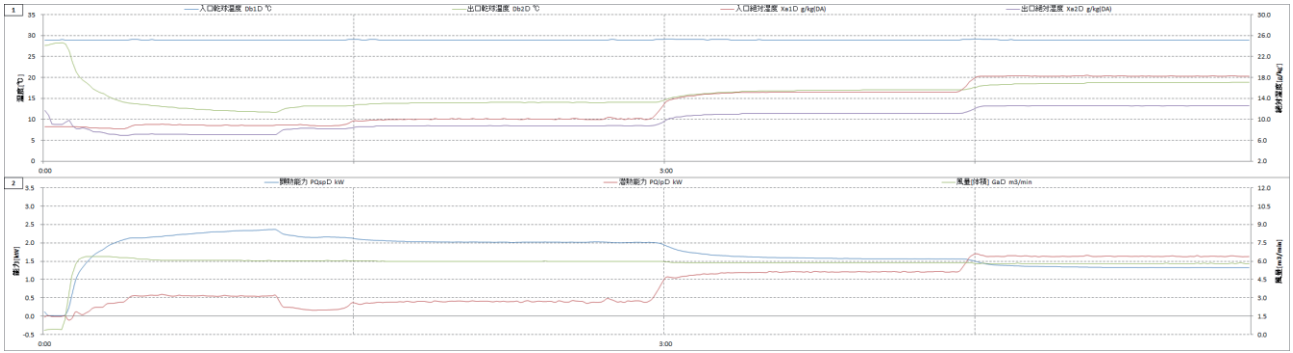


図 93 機種 4 の結果例

(2) 負荷変化実験

負荷変化実験の結果例を図 94 に示す。これは、リモコン設定および顕熱負荷量を固定して、潜熱負荷を変化させた際の機器 1 の実験結果である。単純に目標温度との差で動作していないとみられる動きがあるが、機器制御による影響と考えられる。制御は、メーカーや商品の思想などに依存するため、あくまで今回の実験機種の場合であることに注意が必要である。

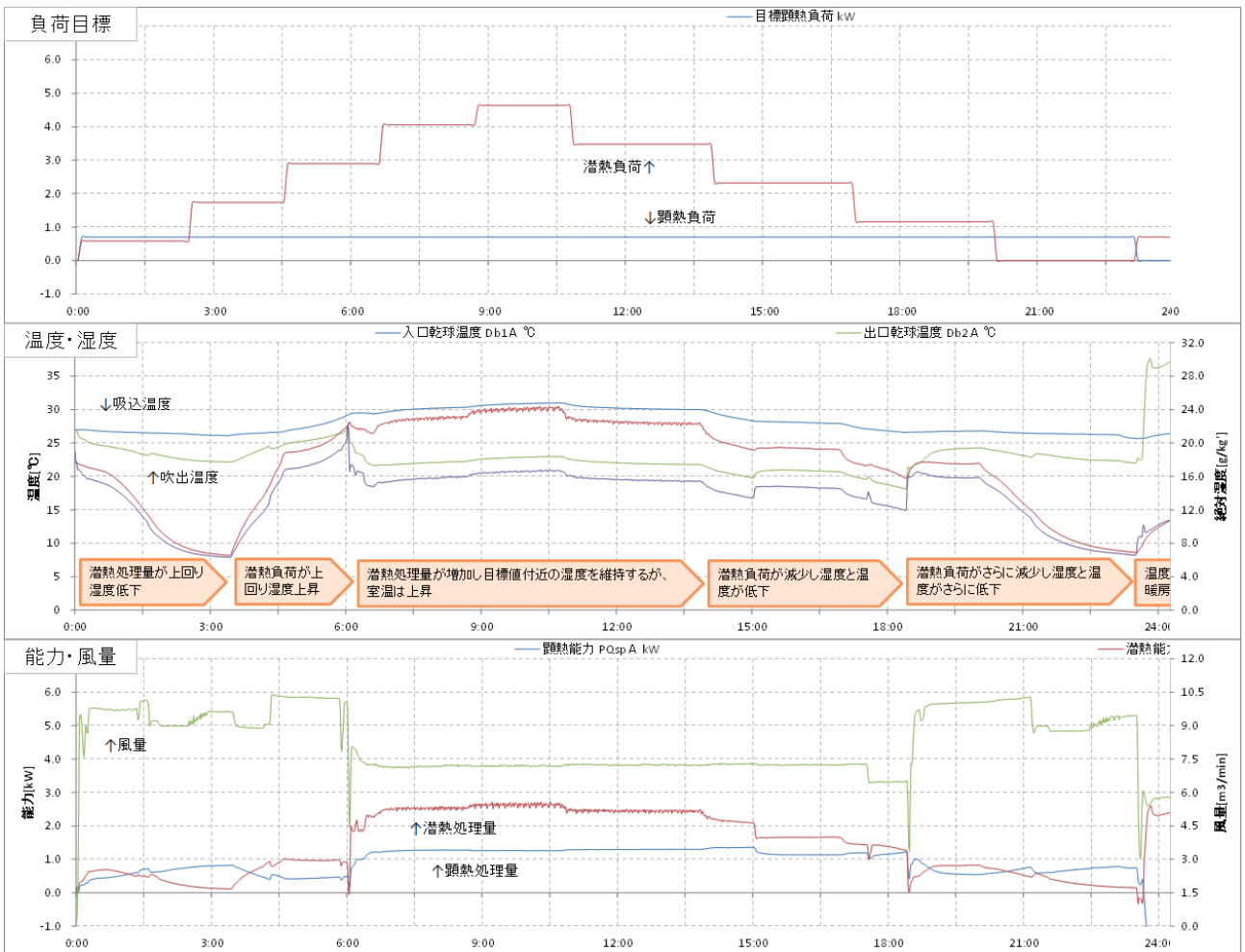


図 94 負荷変化実験結果の例

(3) 室内機内部保水量

空気エンタルピー法による実験結果（温度 27 °C、湿度 47%における冷房運転時）を表 47 に、重量による保水量確認結果を表 48 に示す。一事例ではあるが、多くの水が室内機内部に残っていることがわかる。

表 47 空気エンタルピー法による保水量確認

機器番号	保水量
機種 1	379 g
機種 2	161 g
機種 3	621 g
機種 4	468 g

表 48 重量による保水量確認

機種番号	運転終了後	24 時間後	差
機種 1	14191 g	13970 g	221 g
機種 2	15269 g	14941 g	328 g
機種 3	16205 g	15248 g	958 g
機種 4	17605 g	16591 g	1014 g

(4) フィールド模擬実験

フィールド模擬実験結果について図 95 から図 97 に示す。

実験室実験と同様の結果が得られており、実使用下においても運転中及び停止後において、室内機内部に多くの水が排水されずに貯まっていることが確認できた。

なお、機器運転終了後（停止後）放置した場合は 24 時間でおよそ 400g 程度が放出され、乾燥するまでに 2 日程度を要することが確認できた。また、機器に搭載されている「内部乾燥運転」（機器停止後に一定時間送風を行う運転）を使用した場合には、6 時間で 700g 程度放出されるが、一定時間経過後にも内部は乾ききっていないかった。

これらにより、エアコン内部が高湿度環境となりカビ等の発生が、また室内への再放湿により室内が高湿度になることによる影響が想定される。ただし、水の供給源は除湿によるものであり室内環境に影響を受けるなど、今回確認した 4 機種でも数値にばらつきがあったことなど取り扱いについては注意が必要である

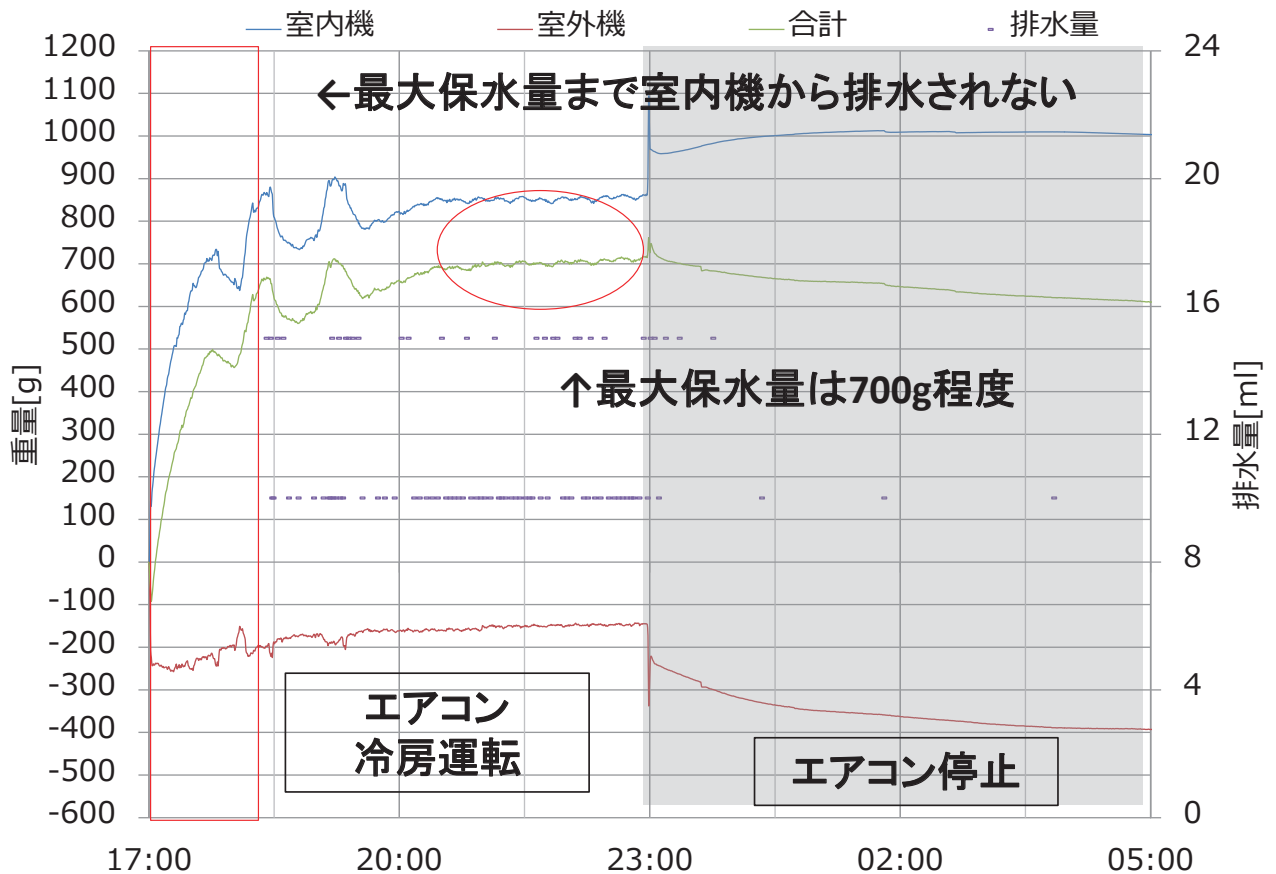


図 95 室内機の保水量と保水速度

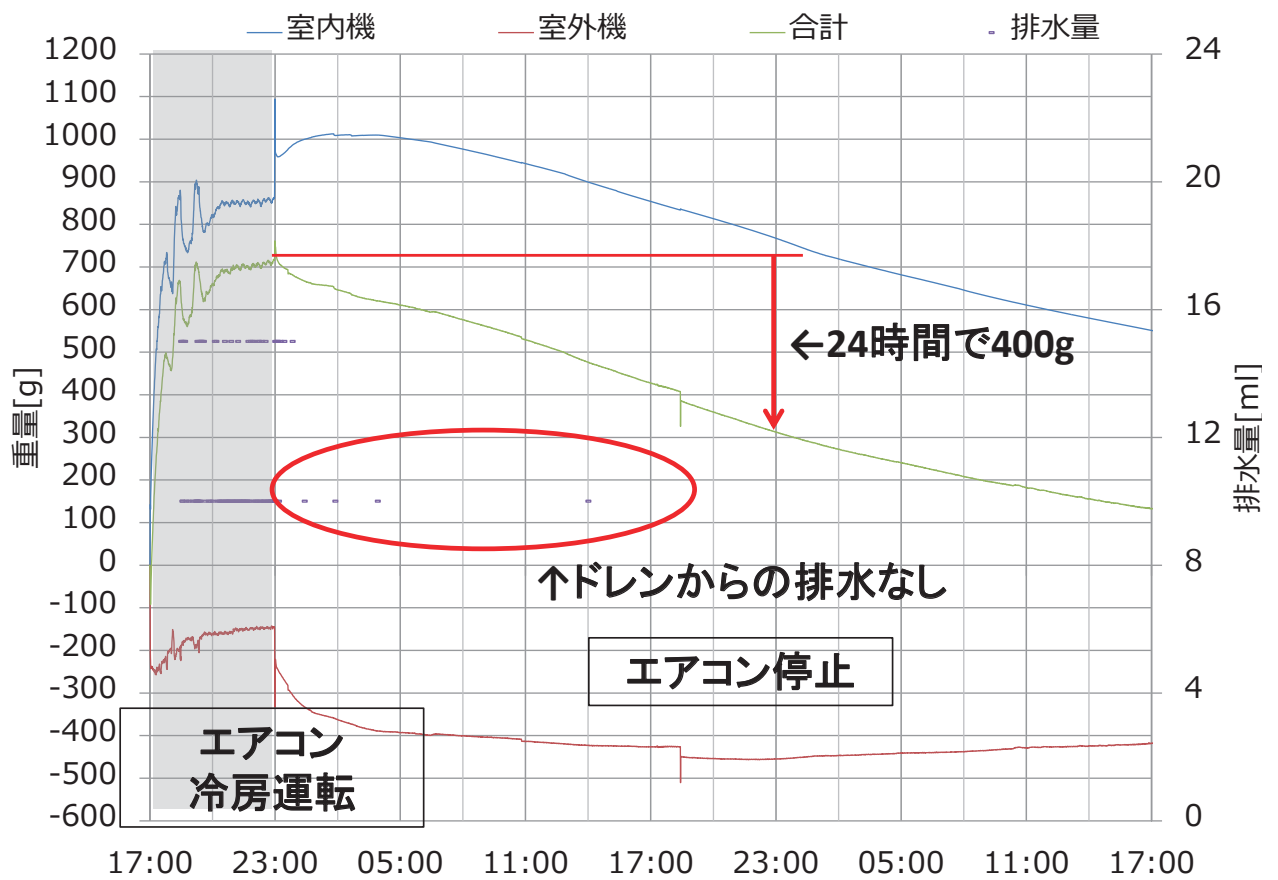


図 96 室内機の放湿速度（内部乾燥運転なし）

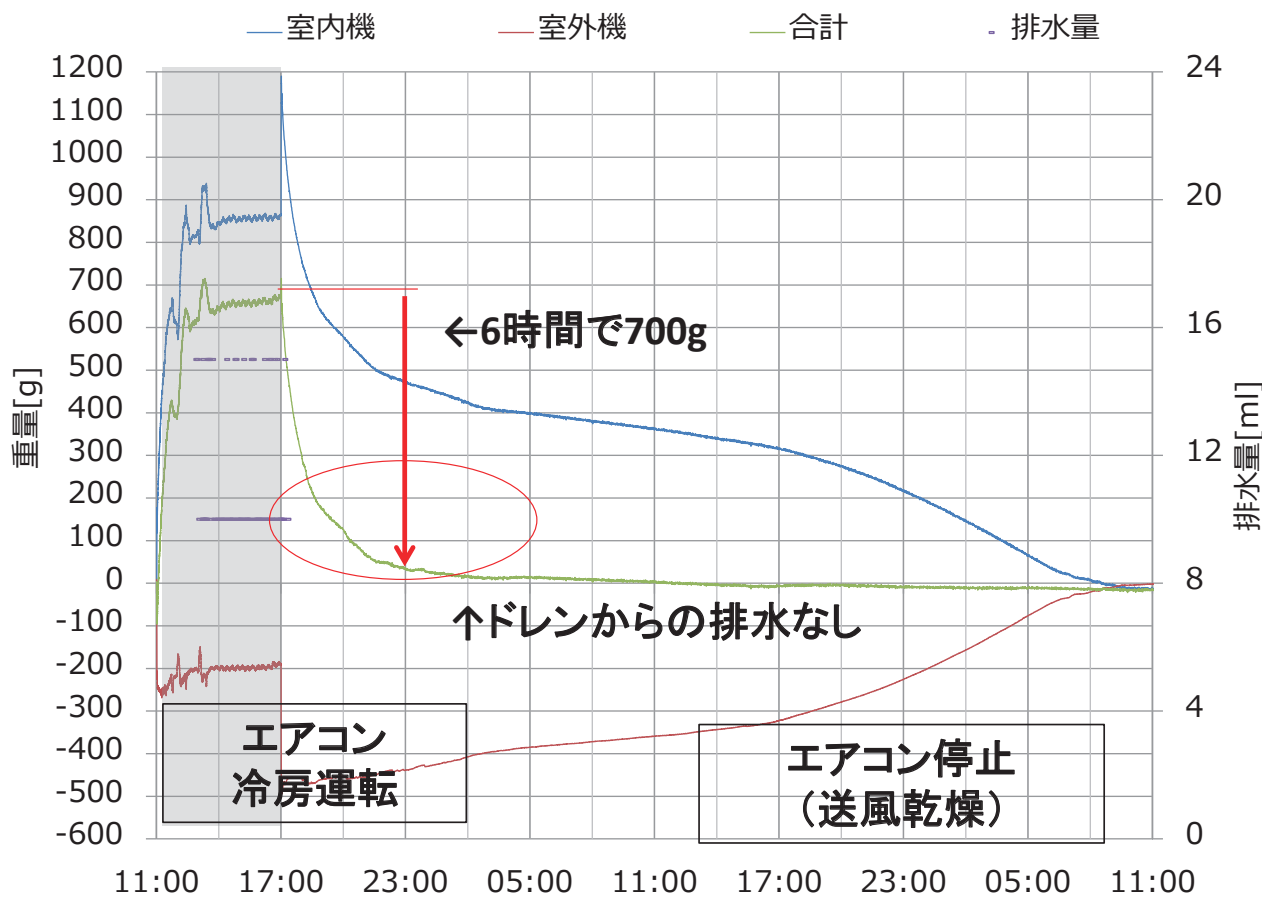


図 97 室内機の放湿速度 (内部乾燥運転あり)

(5) 【参考】ルームエアコンディショナーの風量に室内機熱交換器の結露が及ぼす影響確認

風量低下実験の一例（温度 27℃、湿度 47%における冷房運転時）として時系列データを図 98 に、同条件で実施した際の全機種の結果例を表 49 に示す。

時系列データより、室内機のファン回転数が一定であるが、結露により風量が徐々に減少していることがわかる。また、低下割合の表よりすべての機種において 10%以上の風量低下が共通して発生していることがわかる。

ただし、実験室実験では内部の結露状態（量）がわからないため、参考データとし今後の課題とする。

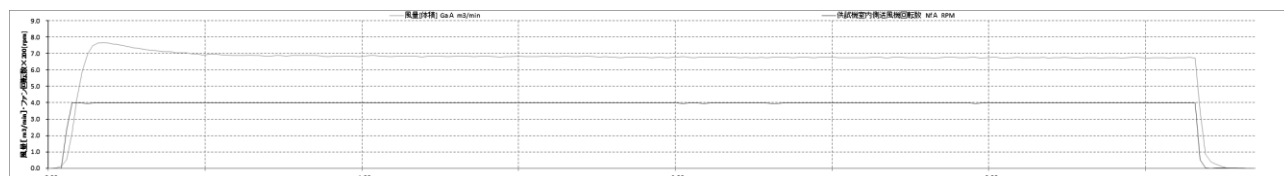


図 98 除湿時の風量低下例

表 49 風量の低下割合例

機種番号	風量 [m ³ /min]			
	max	210 分後	差	割合
機種 1	7.66	6.71	0.95	12%
機種 2	5.29	4.54	0.75	14%
機種 3	6.90	5.83	1.07	16%
機種 4	5.69	4.98	0.71	12%

3.2.3 潜熱処理モデルの作成

3.2.3.1 実験データの解析

取得したデータを用いて、ルームエアコンディショナーの冷房除湿特性について確認した。

一例として、図 99 から図 103 に室内温湿度を 27℃ 60%とした時の冷房運転と除湿運転の違いを示す。冷房運転は主に顕熱を処理し、除湿運転は潜熱を処理するが同時に顕熱も処理する運転であった。

また、この際の室内機熱交換器表面温度は、冷房運転では全体がほぼ一定の温度であったのに対し、除湿運転では熱交換器の温度に分布があった。

除湿運転において処理させる負荷を時間的に変化させた場合に、熱交換器の温度分布がない（冷房と同様の状態）も見られた。

本結果をもとに、代表的な動作状況について空気線図上の動きを図 104～106 に示す。

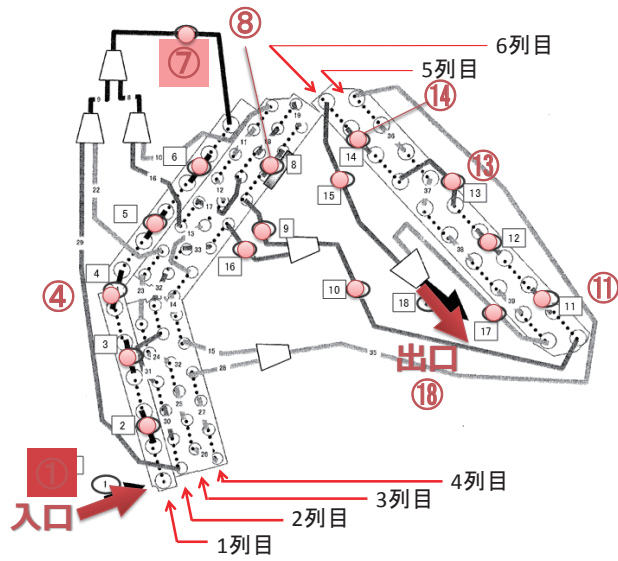


図 99 熱交換器表面の温度計測点

冷房モード (吸込み : 27°C60%)

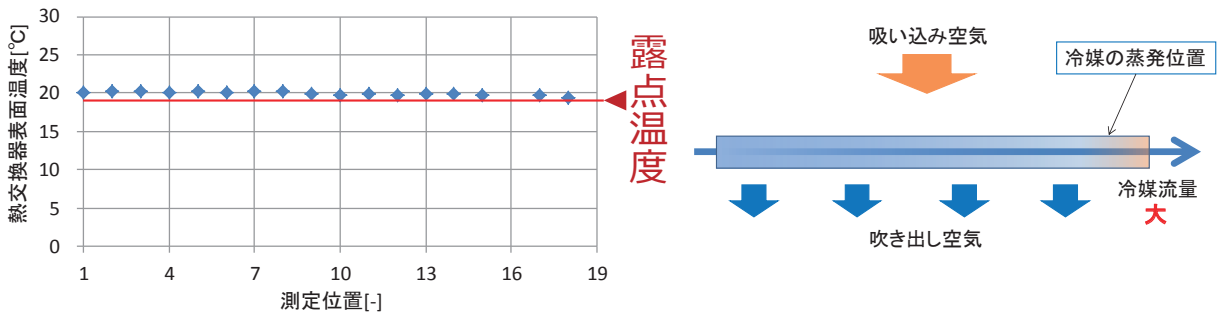


図 100 冷房モード時の熱交換器表面温度

除湿モード (吸込み : 27°C60%)

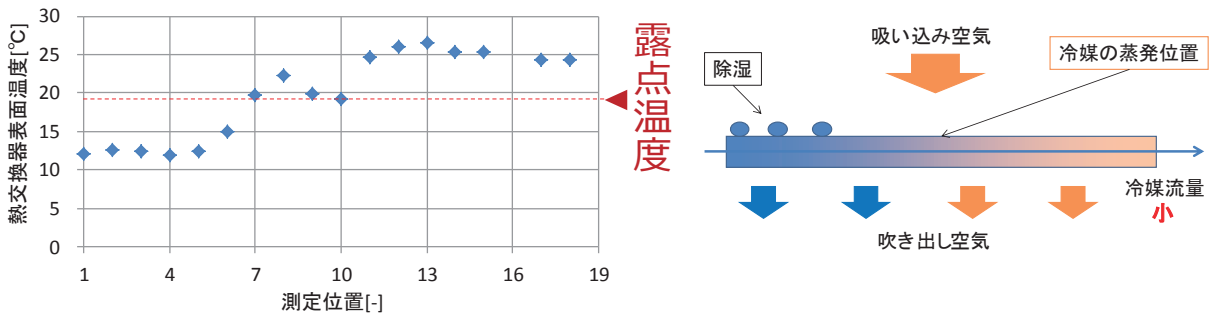


図 101 除湿モード時の熱交換器表面温度

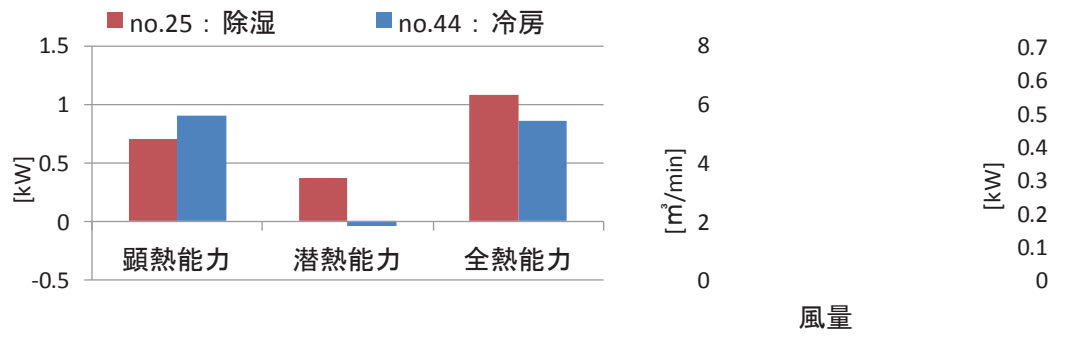


図 102 モードによる処理熱量の比較

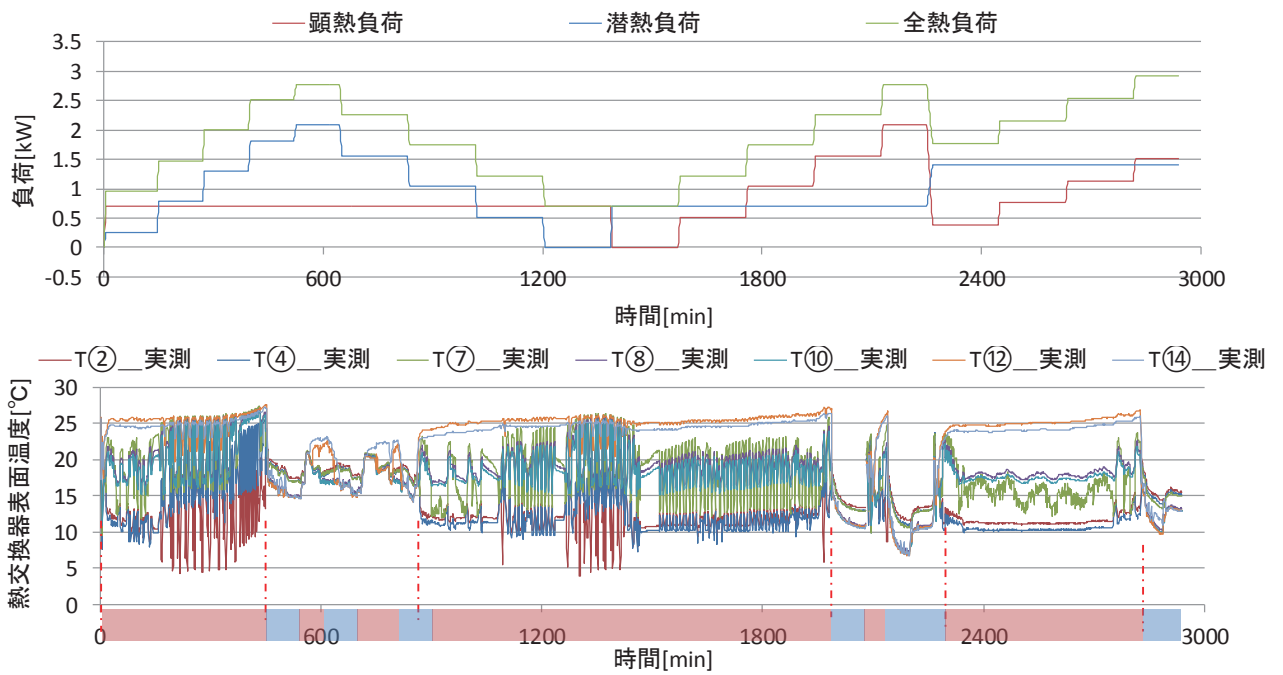


図 103 冷房負荷による熱交換器表面温度の変化

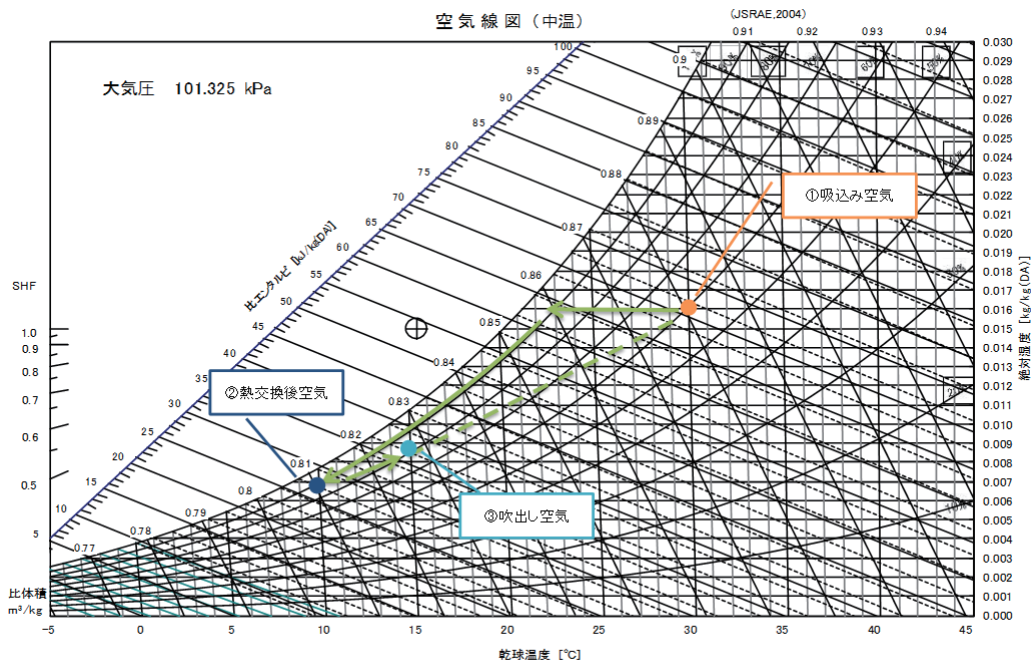


図 104 定格付近の冷房を想定した空気線図例

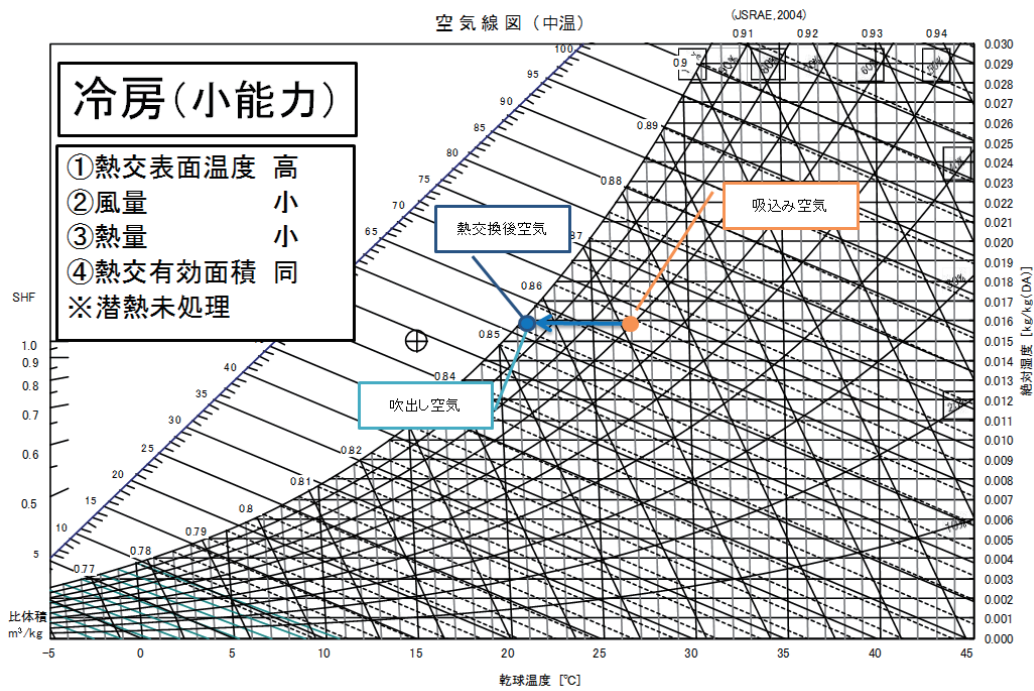


図 105 小能力の冷房を想定した空気線図例

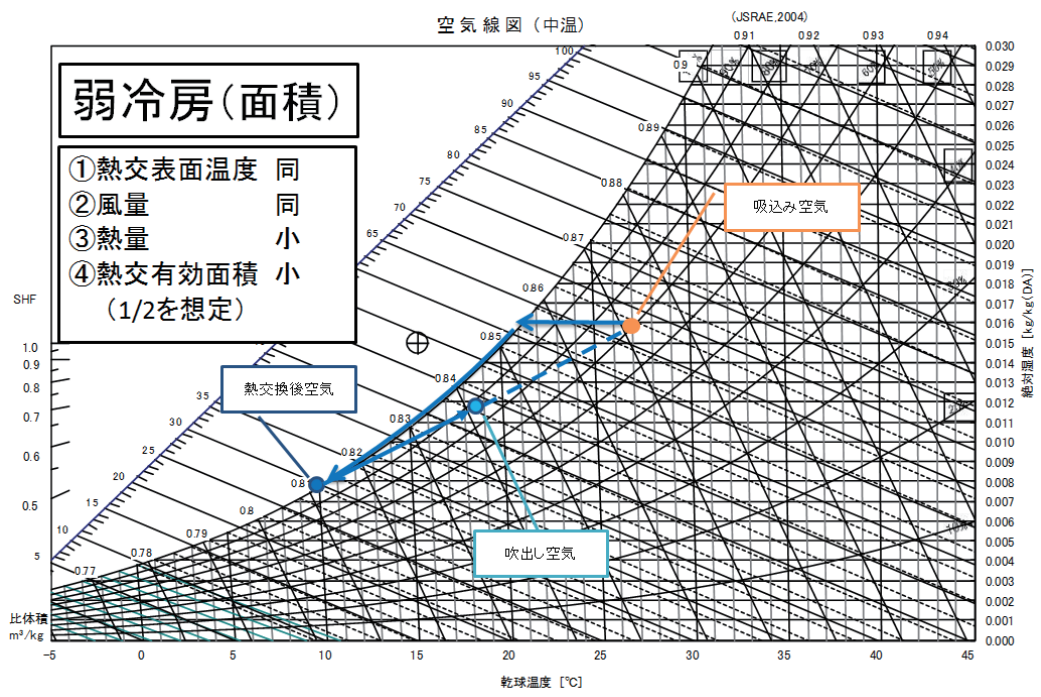


図 106 熱交換器面積による弱冷房除湿を想定した空気線図例

3.2.3.2 除湿モデル因子

実験結果をもとにルームエアコンディショナー特性に関わる因子関連イメージ案を図 107 に示す。

本モデルは、これまでと同様に任意の運転状態におけるエネルギー消費量を出力とするだけでなく、住宅モデルと連携させるための、室内機熱交換器表面温度（ヒートポンプサイクルにおける冷媒蒸発温度）を出力としている。

さらに、これまで検討してきたモデルでは冷房負荷を直接入力（与条件）していたが、住宅モデルと連携させるために、吸込み空気の温湿度ならびに目標温度を入力とする必要がある。

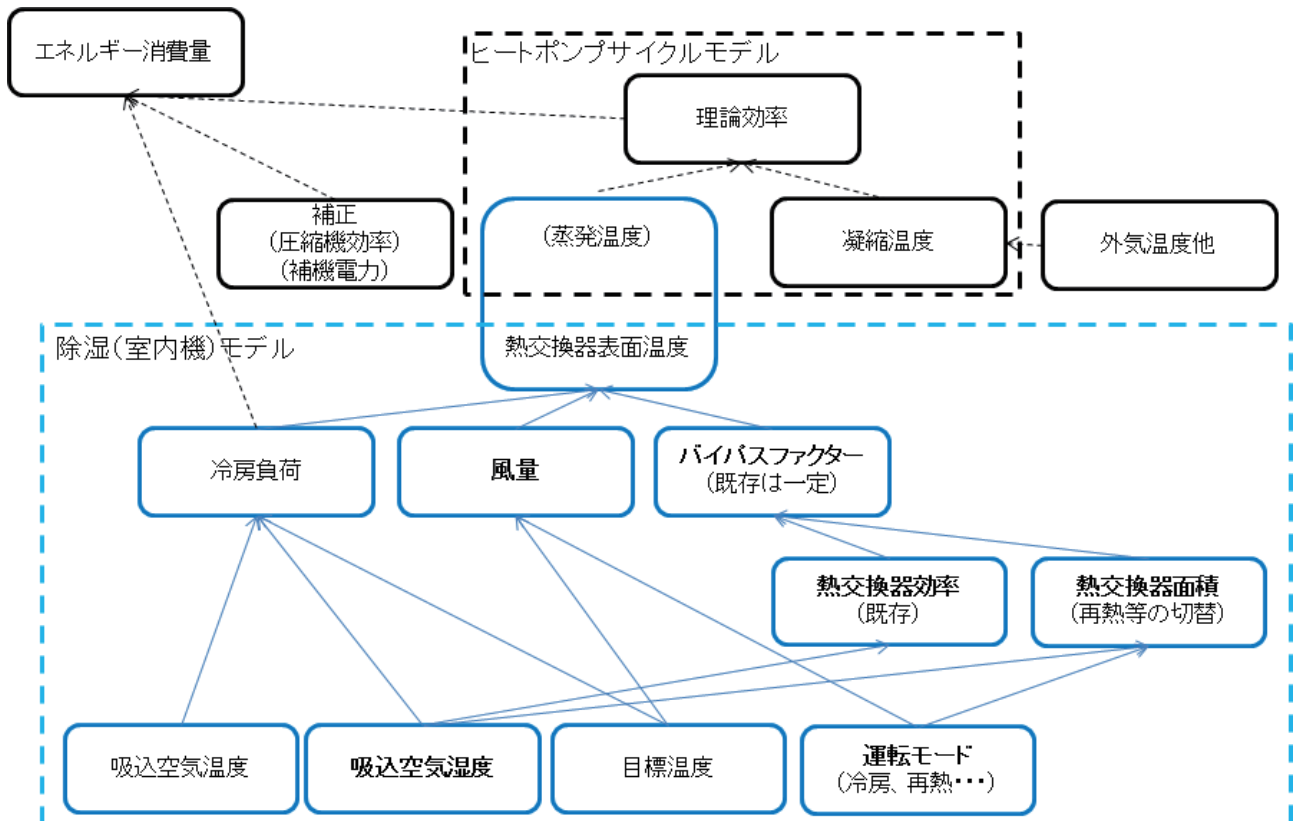


図 107 ルームエアコンディショナー特性に関わる因子関連イメージ

ここで用いるバイパスファクターとは、熱交換器の特性（空気と熱交換器の接触率、熱交換器表面温度と接触空気の温度差など）を包括する補正係数である。

これらの因子の関連を受けより具体的に図 108 に示すような、バイパスファクターを用いたモデルとした。バイパスファクターは、例えば熱交換器の固定や風路により死に面となってしまう比率であったり、熱交換器の熱交換特性であったり複数の意味を持つが、本モデルでは建築分野において従来から用いられてきたバイパスファクターと同意の熱交換器効率を包括するものとして扱った。

なお、熱交換器表面温度と接触空気の温度差を考慮した方がより細かな計算に対応できるが、本モデルでは壁掛けのルームエアコンを対象としており室内機の大きさなどがおおむね共通であり、実際に計算と実測が良く一致したことから、バイパスファクターに包括することとした。

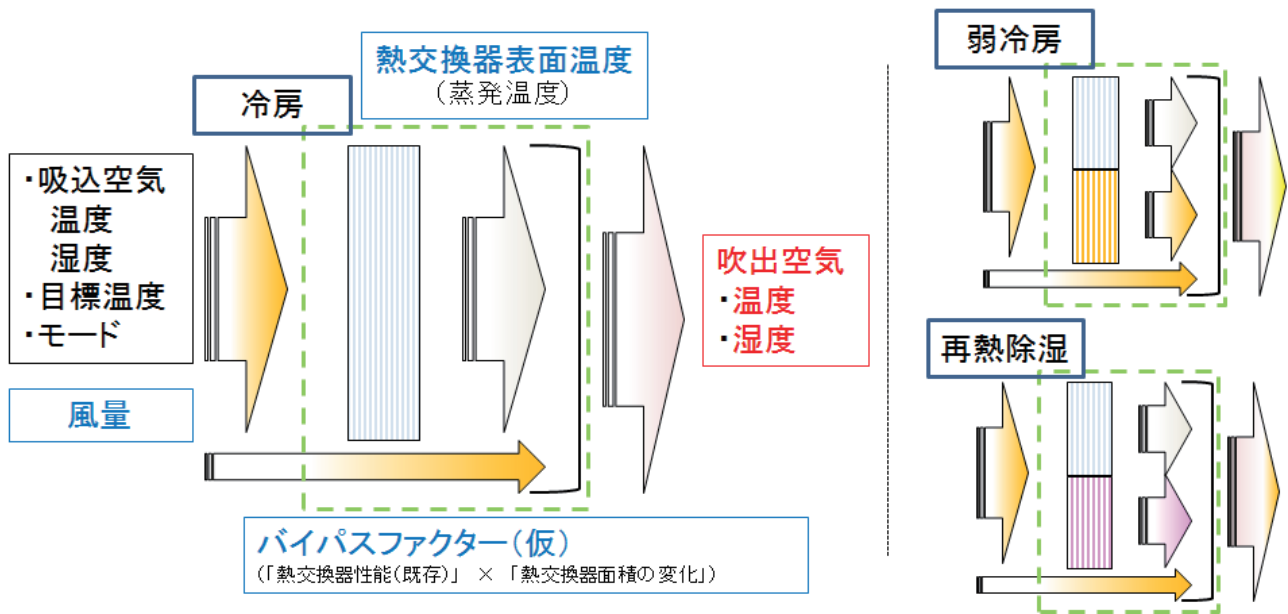


図 108 モデルイメージ

3.2.3.3 計算モデルの作成

実験結果より、冷媒の温度変化を計算することで熱交換器表面温度分布を考慮し、条件に応じた顕熱・潜熱処理量および室内機内部保水量を算出することを目的とした、モデル概要を示す。

(1) モデルの概要

今回構築するモデルの目的は室内側熱交換器の入口温度と冷媒流量を入力値として、冷媒の温度変化を計算し熱交換器表面の温度分布を考慮することで、室内および外気条件に応じた顕熱・潜熱処理量、室内機内部保水量を算定することである。

モデル化にあたり以下の仮定を与える。

- ① 熱交換器でのバイパスファクター (BF) を導入する。
- ② 熱交換器におけるバイパスファクターは一定とする。
- ③ 熱交換器表面温度は冷媒温度に等しいとする
- ④ 熱交換器を二つの領域に分けて、室内機に吸い込まれた空気は一定の割合で分配され、それぞれを通過する空気は途中で混合されない。
- ⑤ 熱交換器の各列での吹き出し空気は完全混合され次の列の吸い込み空気となる。
- ⑥ 熱交換器表面に付着する結露水は上限値を超えると流下し、それ以外は保水される。
- ⑦ 蒸発器では冷媒が気液混合領域において、冷媒が受け取る熱量は相変化にのみ使われ、冷媒の温度変化は圧力低下による温度低下のみを考える。

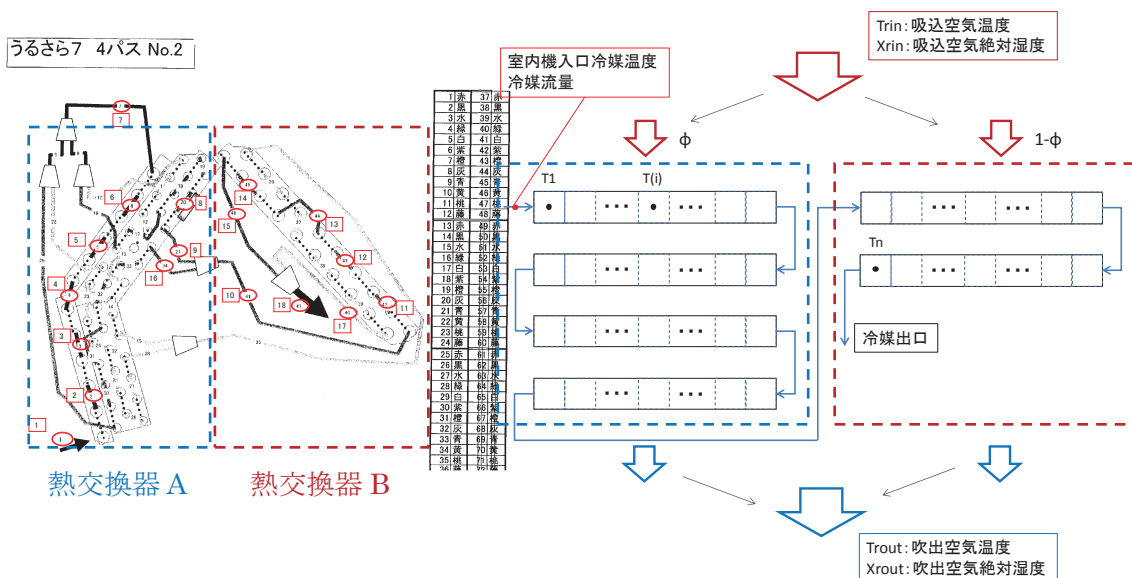


図 109 室内側熱交換器の計算モデル

(2) 基礎理論

(I) 空気側の熱・水分収支モデル

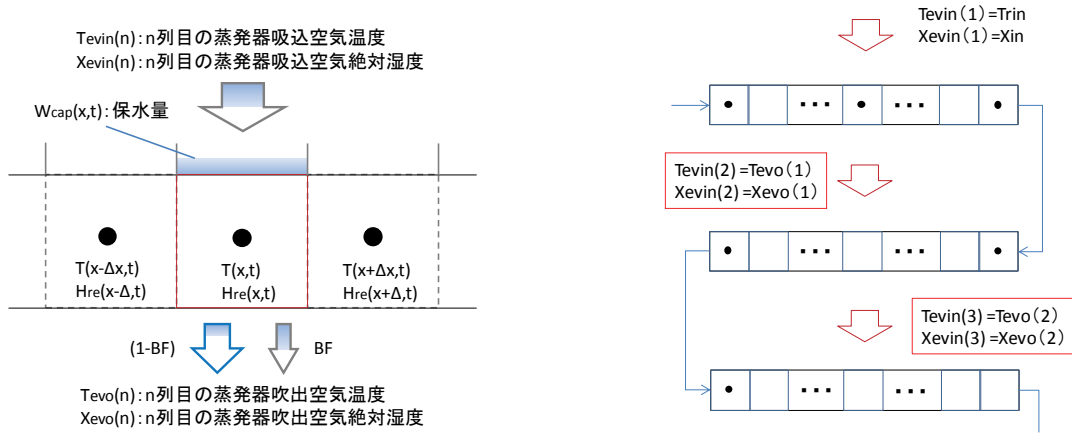


図 110 空気側の計算モデル

(i) 顕熱、潜熱、全熱能力の算定

$$Qsh(x, t) = Mrin(1 - BF) \{ Ca(Tevin - T(x, t)) + Cv(Xevin * Tevin - Xsat(T(x, t))T(x, t)) \}$$

式 7

$$Qlh(x, t) = Mrin * (1 - BF) * Lo * (Xevin - Xsat(x, t))$$

式 8

$$Qth(x, t) = Qsh(x, t) + Qlh(x, t)$$

式 9

(ii) 吹き出し空気温湿度の算定

$$Tev(x, t) = BF * Tevin + (1 - BF) * T(x, t)$$

式 10

$$Xevo(x, t) = BF * Xevin + (1 - BF) * Xsat(T(x, t))$$

式 11

記号

Qsh, Qlh, Qth	: 顕熱、潜熱、全熱処理量	kW
Mrin	: 吸込み空気の質量流量	kg/s
Ca	: 乾燥空気の定圧比熱	kJ/kg · K
Cv	: 水蒸気の定圧比熱	kJ/kg · K
Lo	: 水の蒸発潜熱	kJ/kg
Xsat(T)	: 温度 T に対する飽和絶対湿度	kg/kg'
T	: 熱交換器表面温度	°C
Tevin	: 蒸発器の吸い込み空気温度	°C
Tev	: 蒸発器の吹き出し空気温度	°C
Xevin	: 蒸発器の吸い込み空気絶対湿度	kg/kg'
Xevo	: 蒸発器の吹き出し空気絶対湿度	kg/kg'
BF	: バイパスファクター	-

(iii) 室内機内部保水量の算定

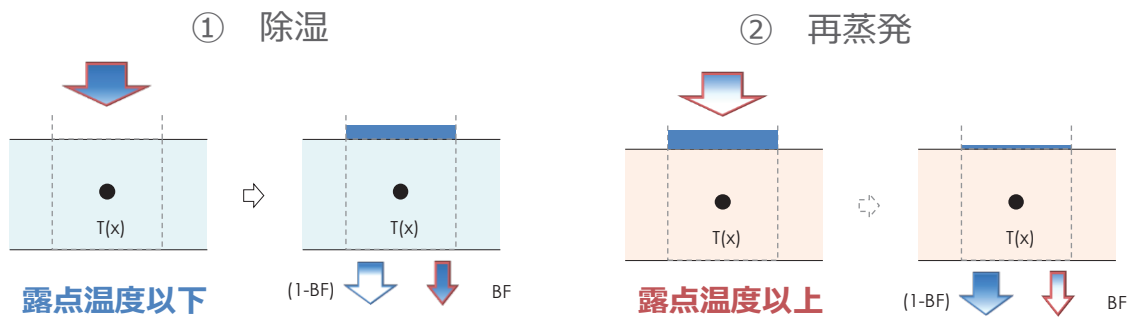


図 111 室内機内部保水量算定モデルの概要

場合分けによる

i) $X_{evin} > X_{sat}(T(x, t))$ かつ $W_{cap}(x, t) < W_{capmax}$: 除湿されるかつ最大保水量以下

$$\frac{\partial W_{cap}}{\partial t} = Mrin * (1 - BF) * \{X_{evin} - X_{sat}(T(x, t))\}$$

式 12

$$W_{cap}(i, t + 1) = W_{cap}(i, t) + Mrin * (1 - BF) * \{X_{evin} - X_{sat}(T(x, t))\} * \Delta t$$

式 13

ii) $X_{evin} > X_{sat}(T(x, t))$ かつ $W_{cap}(x, t) = W_{capmax}$: 除湿されるかつ最大保水量

$$W_{cap}(x, t + \Delta t) = W_{capmax}$$

式 14

iii) $X_{evin} \leq X_{sat}(T(x, t))$ かつ $W_{cap}(x, t) > 0$: 除湿されないかつ保水あり → 再放湿

$$W_{cap}(x, t + \Delta t) = W_{cap}(i, t) + Mrin * (1 - BF) * \{X_{evin} - X_{sat}(T(x, t))\} * \Delta t$$

式 15

iv) $X_{evin} \leq X_{sat}(T(x, t))$ かつ $W_{cap}(x, t) = 0$: 除湿されないかつ保水なし

$$W_{cap}(x, t + \Delta t) = 0$$

式 16

記号

W_{cap} : 室内機内部保水量 kg

(II) 冷媒側の熱・水分収支モデル

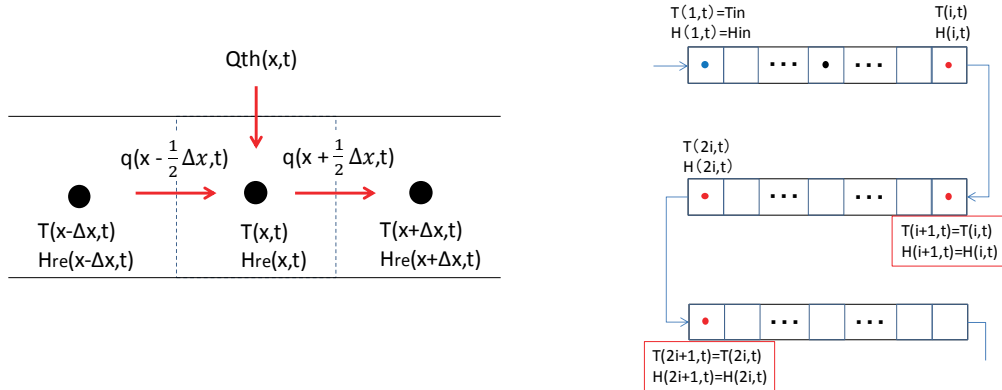


図 112 冷媒の計算モデル

(i) 熱収支式

$$\rho C \frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x} + Q_{th}(x, t)$$

式 17

ここで、定常時には左辺が 0 なので

$$H_{re}(x + \Delta x, t) = H_{re}(x, t) + \frac{Q_{th}(x, t)}{m_{re}}$$

式 18

(ii) 冷媒の温度変化について

i) $H_{re}(x + \Delta x, t) < H_{ev}$ 冷媒が気液混合時：仮定⑦より次式となる

$$T(x + \Delta x, t) = T(x, t) - \Delta T_p$$

式 19

ii) $H_{re}(x, t) < H_{ev} < H_{re}(x + \Delta x, t)$ 冷媒が途中で蒸発しきる時

$$T(x + \Delta x, t) = T(x, t) + \frac{Q_{th}(x, t) - Q_{relh}(x, t)}{m_{reg} C_g} - \Delta T_p$$

式 20

$$Q_{relh}(x, t) = (H_{ev} - H_{re}(x, t)) * m_{rel}$$

式 21

iii) $H_{ev} < H_{re}(i)$ 冷媒が蒸発しきった後

$$T(x + \Delta x, t) = T(x, t) + \frac{Q_{th}(x, t)}{m_{reg} C_g} - \Delta T_p$$

式 22

記号

Hre	: 冷媒の比エンタルピー	kJ/kg
mre	: 冷媒流量	kg/s
ΔT_p	: 圧力損失による温度低下	°C

(III) 冷媒の圧力低下に伴う温度低下について

冷媒は気液混合領域では相変化により熱交換を行うため、理想的には温度変化は起きないが、実際には温度は一定ではなく、熱交換器の後半になるほど温度が低くなっている。この傾向は能力が大きくなるほど顕著であった。これは配管での圧力損失によって冷媒圧力が低下することが原因であると考えられる。そこで以下のように圧力損失による温度低下を推定する。

式 24 を用いて冷媒の入口温度から冷媒圧力を求める。冷媒の圧力低下は摩擦による圧力損失であるとし、式 23 を用いて計算する。式 26 を用いて冷媒圧力から冷媒温度を計算する。

$$\Delta P = \lambda * \frac{l}{d} * \frac{\rho}{2} V^2$$

式 23

$$P(x, t) = 0.000002 * T(x, t)^3 + 0.0003 * T(x, t)^2 + 0.0254 * T(x, t) + 0.7989$$

式 24

$$P(x + \Delta x, t) = P(x, t) - \Delta P$$

式 25

$$T(x + \Delta x, t) = -2.8348P(x + \Delta x, t)^4 + 22.459P(x + \Delta x, t)^3 - 67.426P(x + \Delta x, t)^2 + 111.63P(x + \Delta x, t) - 56.029$$

式 26

$$\Delta T_p = T(x + \Delta x, t) - T(x, t)$$

式 27

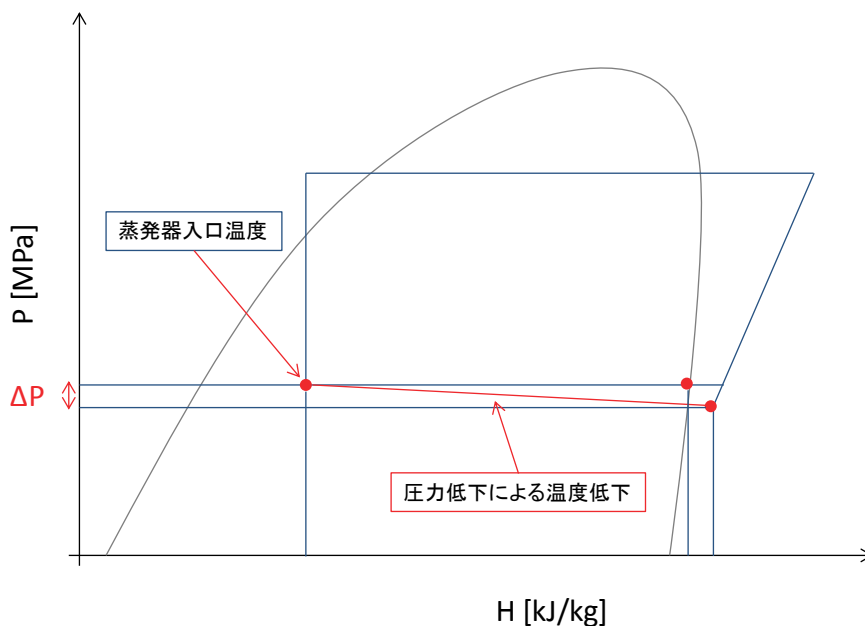


図 113 圧力損失による温度低下のイメージ図

3.2.4 モデルの検証

機種 4 を対象として詳細なモデル検討を行った。図 114 のように熱交換器の列数を考慮したモデル化を行う。対象機器の構造に合わせて熱交換器 A を 4 列、B を 2 列とした。

実測結果と解析結果の比較によりモデルの妥当性を検証する。モデルの検証にあたり入力値として表 50 入力条件の値を用いる。弱冷房除湿モデルでは冷媒流量および蒸発器の入口における冷媒の比エンタルピーが不明であり、再熱除湿モデル冷媒流量、再熱器入口の冷媒比エンタルピー、冷却器入口の冷媒比エンタルピーが不明である。

そこでそれらについては実測値を用いた計算により決定した。

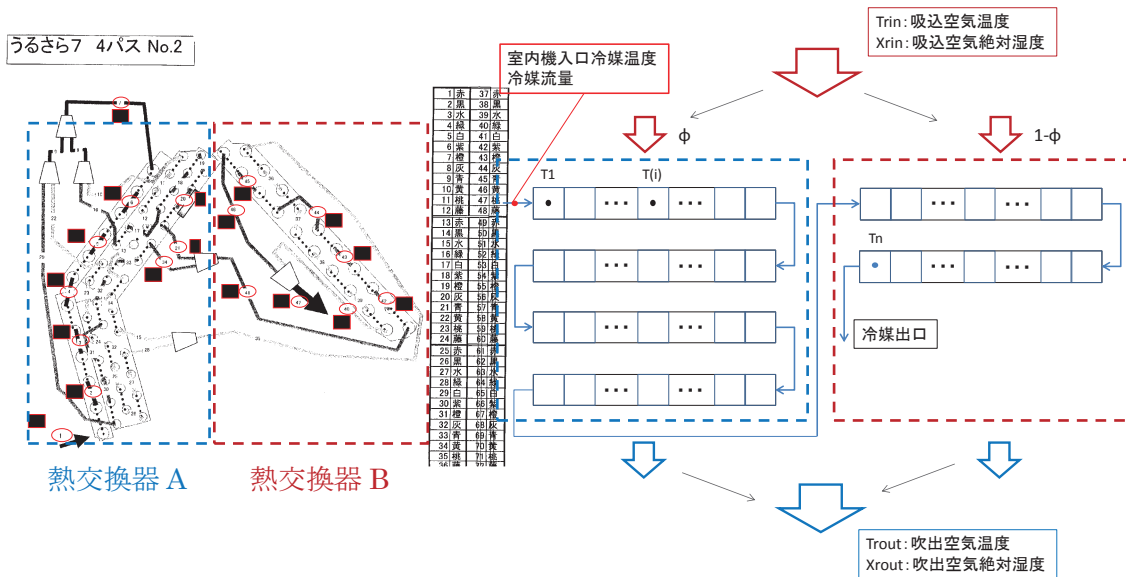


図 114 弱冷房除湿機の計算モデル

表 50 入力条件

Trin	°C	吸い込み空気温度	実測値
Xrin	kg/kg'	吸い込み空気絶対湿度	実測値
Tout	°C	外気温度	実測値
Xout	kg/kg'	外気絶対湿度	実測値
Texpin	°C	膨張弁入口温度	実測値
V	m ³ /min	風量	実測値
Tin	°C	冷媒入口温度	実測値
Hin	kJ/kg	冷媒入口比エンタルピー	計算
Mre	g/s	冷媒質量流量	計算
BF	-	バイパスファクター	仮定
φ	-	風量の流入割合	仮定

3.2.4.1 弱冷房除湿モデルの解析結果

(I) 温湿度固定実験

温湿度固定実験の結果より冷房モード、除湿モードの中で運転が安定した全 82 条件について解析値と実測値の結果の比較を行いモデルの妥当性を検証する。入力値と出力値を表 51 入力値、表 52 に示す。

表 51 入力値

Trin	°C	吸い込み空気温度	実測値	25~28.97
RHrin	%	吸い込み相対絶対湿度	実測値	35~90
Tout	°C	外気温度	実測値	冷房 35、除湿 27
RHout	%	外気相対湿度	実測値	冷房 40、除湿 60
V	m ³ /min	風量	実測値	0.24~9.18
Tin	°C	冷媒蒸発器入口温度	実測値	7.24~26.99
Hin	kJ/kg	冷媒入口比エンタルピー	計算	
Mre	g/s	冷媒質量流量	計算	
BF	-	バイパスファクター	仮定	0.15
φ	-	風量の流入割合	仮定	0.55

表 52 出力値

Trout	°C	吹き出し空気温度		
Xrout	kg/kg'	吹き出し空気絶対湿度		
T	°C	熱交換器表面温度		
Qcl	kW	冷房能力		
Qsh	kW	顕熱能力		
Qcl	kW	潜熱能力		
Wcap	g	室内機内部保水量		

(II) 熱交換器表面温度

図 115 に出力点と計測点の対応関係を示す。今回の計算では各列 200 分割、計 1200 分割した計算を行った。出力した点は 1、50、100、150、200 のように 50 刻みで行った。

熱交換器表面温度の計算値と実測値の比較を図 116~118 に示す。2~3 列目については実測値がないため不明であるが、その他の部分については計算と実測の結果は大略再現されている。除湿モードでは冷媒は 1 列目の途中で蒸発しきり温度が上がり、2 列目では 1 列目の吹き出し温度との熱交換により温度が決まる。3、4 列目についても同様であり 5 列目は室内温度に近づく。

冷房モードでは列の後半なるほど温度が下がっており、圧力低下を考慮することで再現出来た。また、吸込み温度によらず熱交換器表面温度は大略再現された。

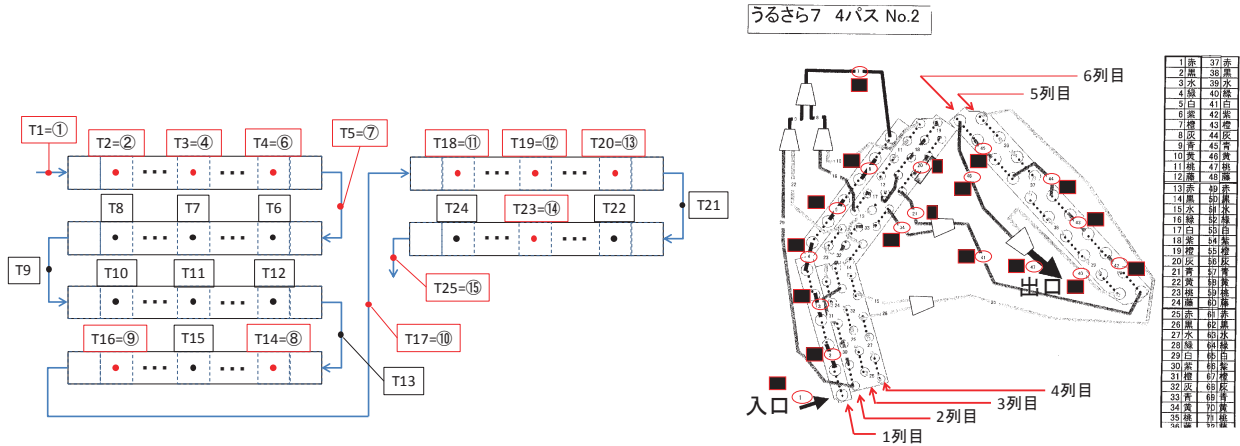


図 115 熱交換器表面温度出力値と計測点の対応関係

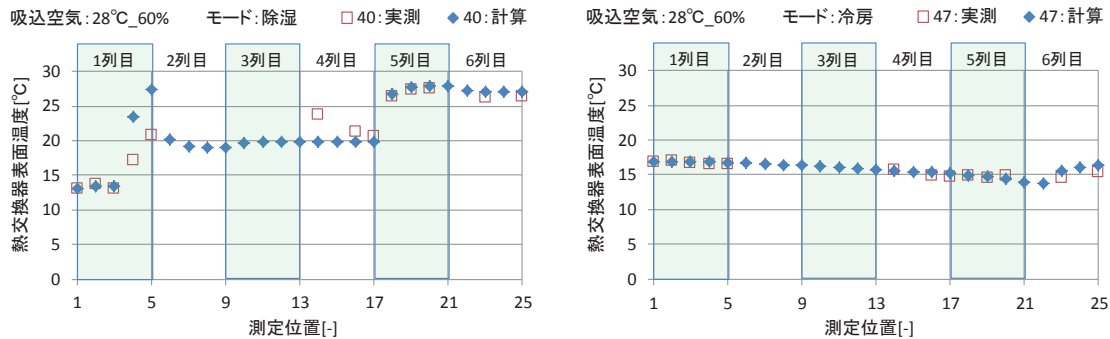


図 116 熱交換器表面温度比較 (28°C60%) 左：除湿、右：冷房

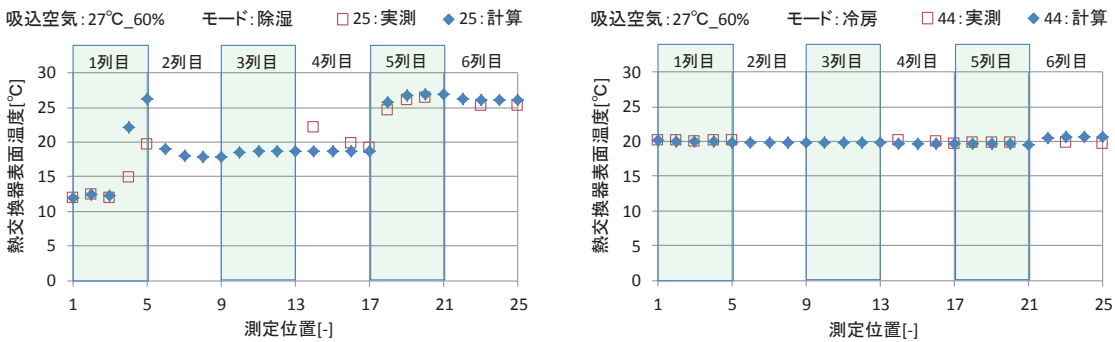


図 117 熱交換器表面温度比較 (27°C60%) 左：除湿、右：冷房

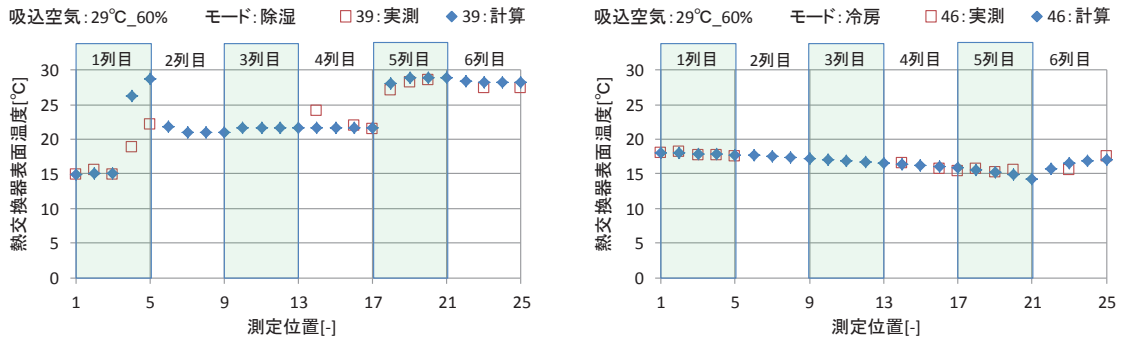


図 118 熱交換器表面温度比較 (29°C60%)

(III) 吹出空気温湿度

図 119 に吹き出し空気の乾球温度と絶対湿度の実測値と計算値の比較結果を風量ごとに色分けして示す。

風量が 1 以下のように非常に小さい場合には乾球温度、絶対湿度ともにずれが大きくなる傾向にあるが、それ以外の領域では計算値と実測値はよく一致した。

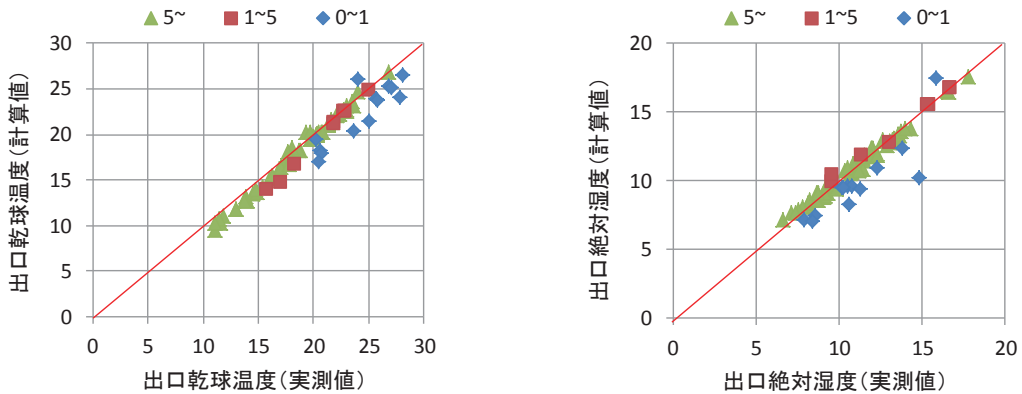


図 119 吹出空気温湿度比較 左：乾球温度、右：絶対湿度

(IV) 顕熱・潜熱能力

図 120 に顕熱能力・潜熱能力の実測値と計算値の比較結果を風量ごとに色分けして示す。また、実施した温湿度固定実験の条件を表 53 に示す。

顕熱・潜熱能力は全領域において計算値と実測値はよく一致しており、上記結果より、モデルの妥当性が示された。

表 53 実験条件

室内機設定モード	冷房or除湿	室内機設定温度	27℃
室内機設定湿度	除湿：50%	風量	自動
外気温度	冷房35℃、除湿27℃	外気相対湿度	冷房40%、除湿60%
室内温度	13~29℃	室内相対湿度	35~90%

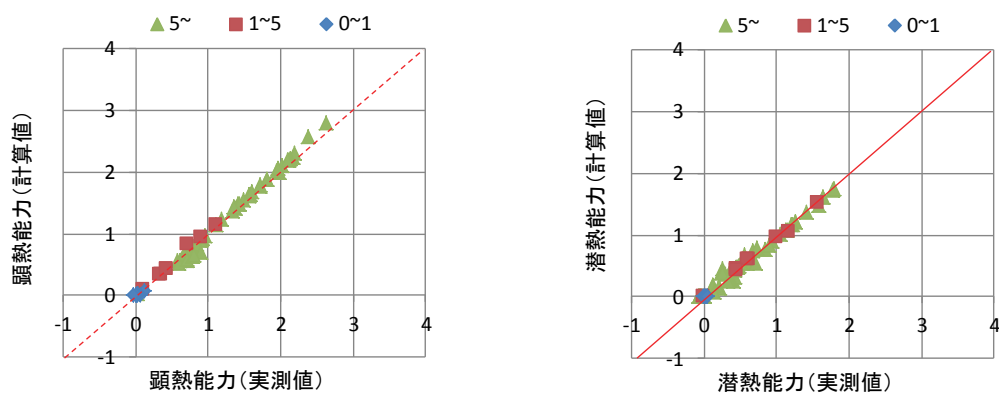


図 120 顕熱・潜熱能力比較 左：顕熱能力、右：潜熱能力

(2) モデルを用いた室内機内部保水量の解析

熱交換器の場所によって温度が異なるため場所によって除湿量が異なると考えられる。そこで各出力点における除湿量を図 121 から図 123 に示す。

除湿モードでは熱交換器の 1 列目でのみ除湿されているのに対し、冷房モードでは熱交換器全体で除湿していることが分かる。また 1 列目で除湿された空気が 2 列目の吸い込み空気となるため除湿量は 1 よりも 2 列目、2 よりも 3 列目といったように減少していることが分かる。

また吸込空気温度が 29°C の時は 28°C と同様の傾向を示した。27°C の時は冷房モードでは熱交換器表面が露点温度以下にならず除湿されていないことが分かる。

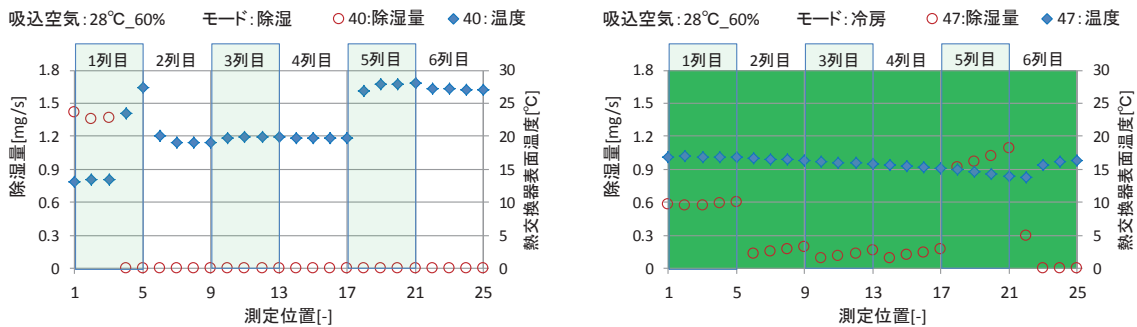


図 121 除湿量分布 (28°C60%) 左: 除湿、右: 冷房

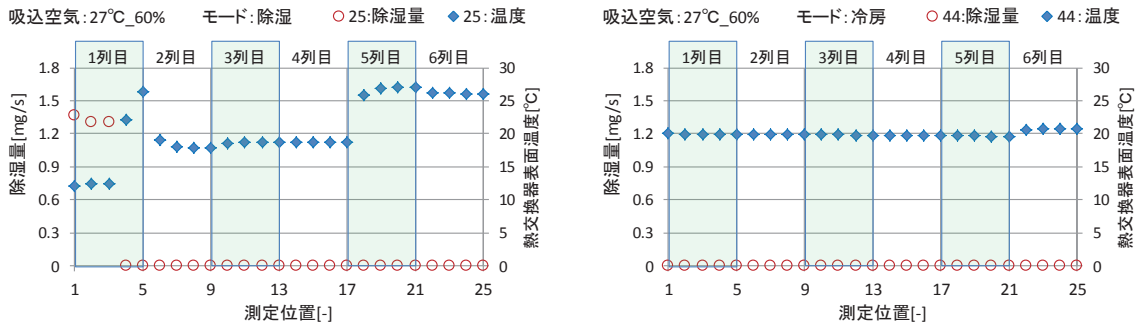


図 122 除湿量分布 (27°C60%) 左: 除湿、右: 冷房

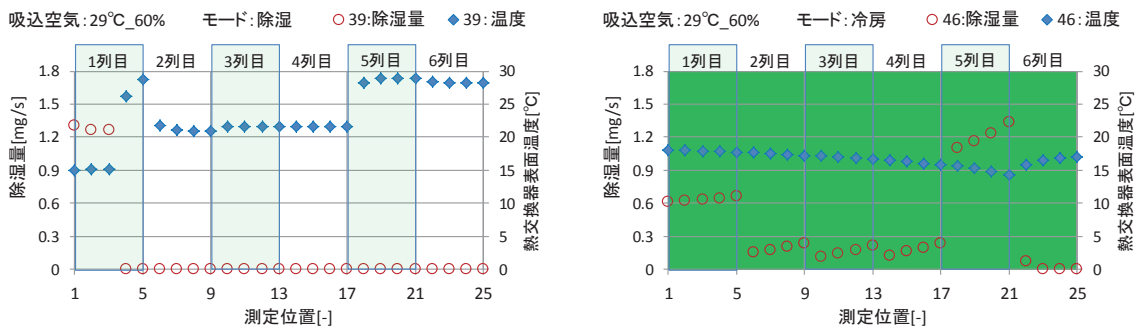


図 123 除湿量分布 (29°C60%) 左: 除湿、右: 冷房

3.2.4.2 負荷固定実験

負荷固定実験の結果について計算値と解析値の比較を示す。表 54 表 55 に入力値と出力値を示す。

表 54 入力値

Trin	°C	吸い込み空気温度	実測値	
RHrin	%	吸い込み相対絶対湿度	実測値	
Tout	°C	外気温度	実測値	27
RHout	%	外気相対湿度	実測値	60
V	m ³ /min	風量	実測値	
Tin	°C	冷媒蒸発器入口温度	実測値	
Hin	kJ/kg	冷媒入口比エンタルピー	計算	
Mre	g/s	冷媒質量流量	計算	
BF	-	バイパスファクター	仮定	0.15
φ	-	風量の流入割合	仮定	0.55

表 55 出力値

Trout	°C	吹き出し空気温度		
Xrout	kg/kg'	吹き出し空気絶対湿度		
T	°C	熱交換器表面温度		
Qcl	kW	冷房能力		
Qsh	kW	顕熱能力		
Qcl	kW	潜熱能力		
Wcap	g	室内機内部保水量		

(I) 熱交換器表面温度

熱交換器の表面温度の比較を図 124 に、熱交換器表面温度分布の実測値（上）と計算値（下）の比較を図 125 に示す。

測定点⑦でずれがみられるものの他の部分については全範囲について実測値と計算値はよく一致している。

冷房除湿時の冷媒の挙動をよく再現されている。

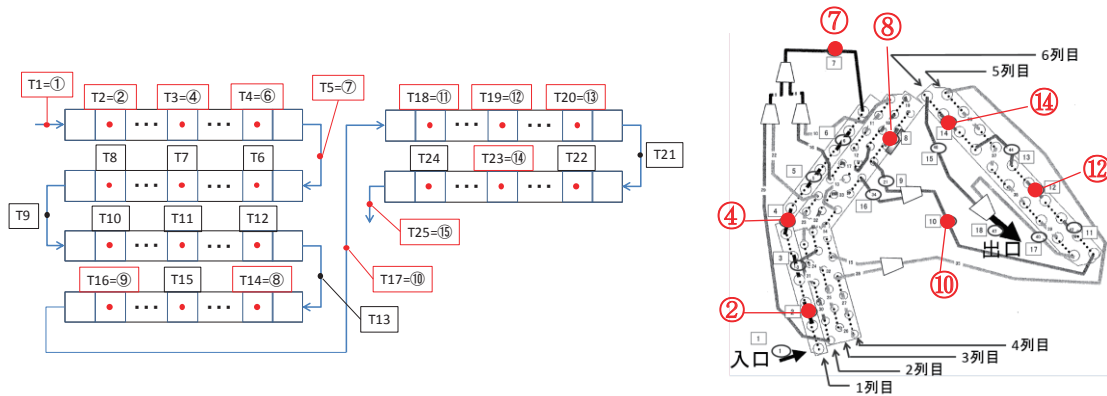


図 124 熱交換器表面温度出力値および計測点との対応関係

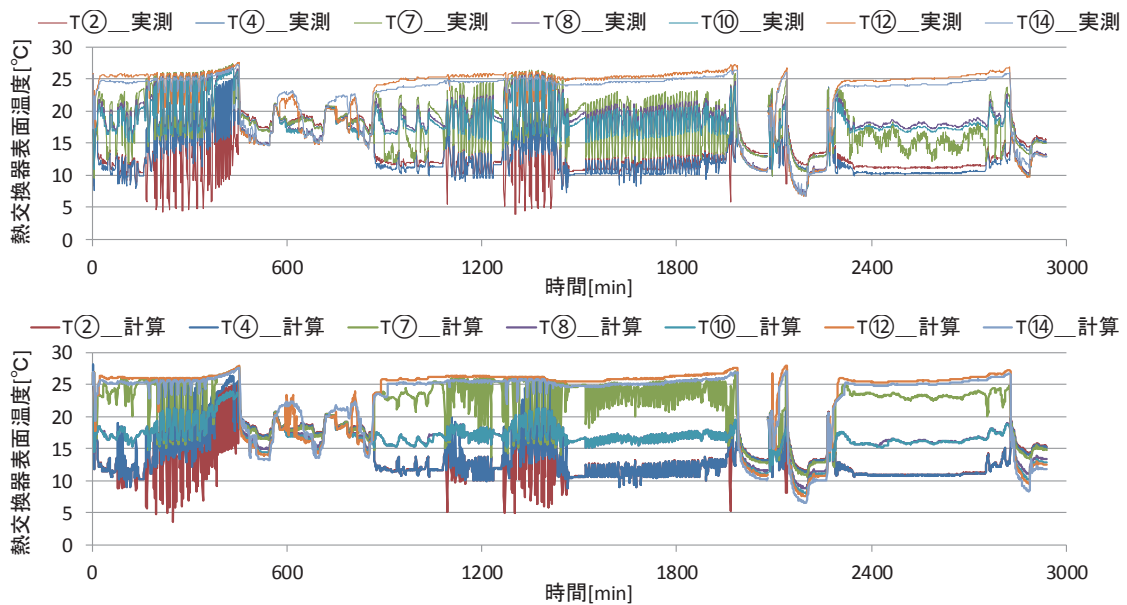


図 125 熱交換器表面温度比較 上：実測値、下：計算値

(II) 吹き出し空気乾球温度、絶対湿度

図 126 に吹き出し空気温湿度の計算値と実測値の比較を示す。

冷房除湿時の吹き出し温度で最大 4℃程度のずれが見られるが、吹き出しの乾球温度、絶対湿度ともに大略再現されているといえる。

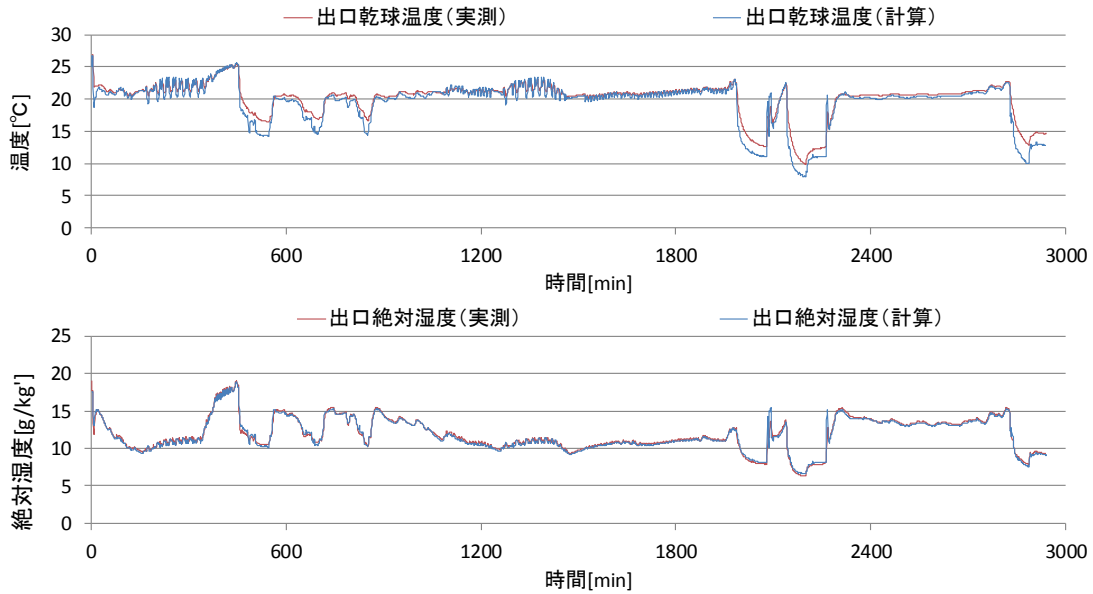
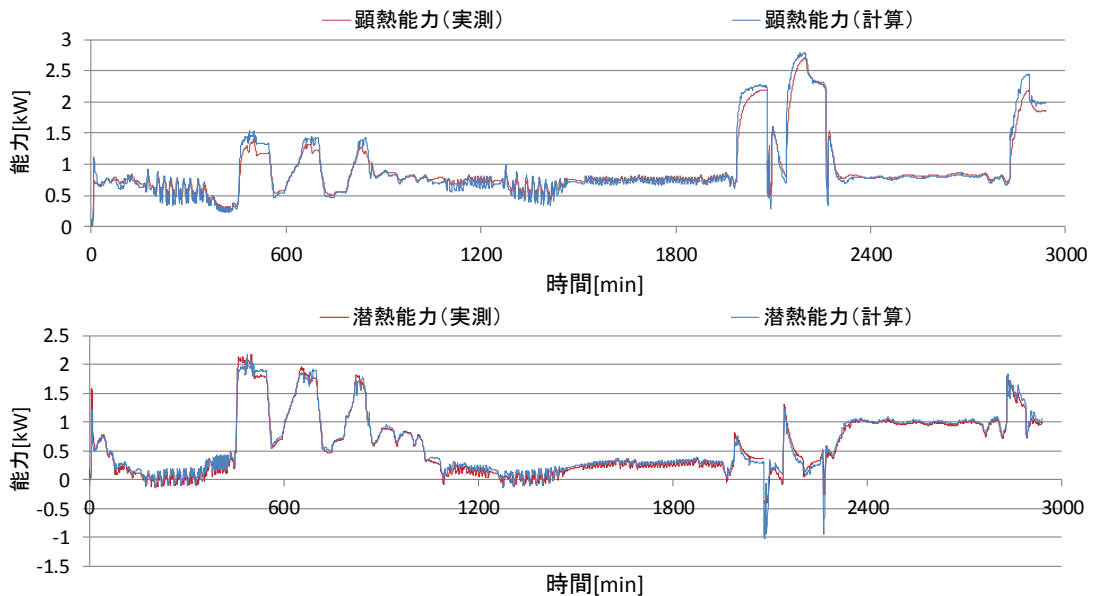


図 126 吹き出し空気 上：乾球温度比較、下：絶対湿度比較

3.2.4.3 顕熱・潜熱能力

顕熱、潜熱能力はよく再現された。



(1)

図 127 能力の比較 上：顕熱能力比較、下：潜熱能力比較

3.2.4.4 【参考】モデルによる室内機内部保水量の推定

計算により室内機内部保水量の推定を行った。保水量の最大値は保水量確認実験より得られた値を用いた。

弱冷房除湿時には冷却面積が小さく除湿可能部分が少ないため、内部保水量は少ないが、冷房除湿では熱交換器全体で冷却・除湿を行うために内部保水量も大きくなっている。

運転停止後の室内機からの再放湿は冷房除湿終了後に大きくなると考えられる。

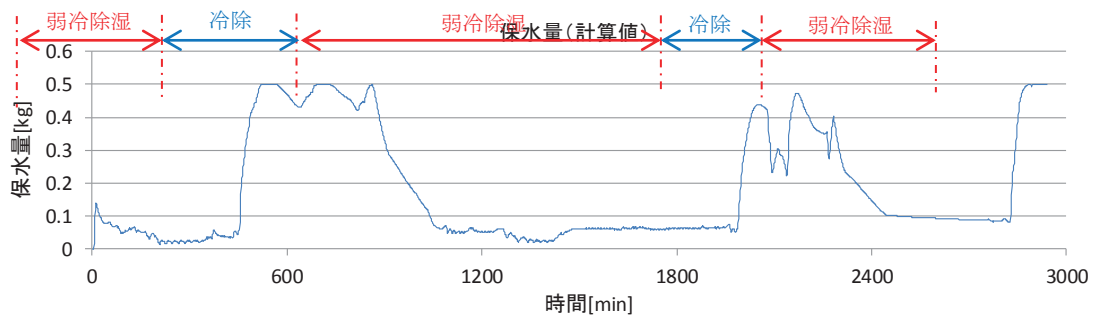


図 128 室内機内部保水量 (計算値)

3.2.5 計算プログラムへの実装

作成したモデルは、室内機熱交換を細分化して、さらに繰り返し計算をするため計算量が非常に多い。そこで、省エネ法の Web プログラムで計算できる程度に、計算量を削減することを目的として、冷房モデルの簡易化を行った。

詳細計算結果を用いて、室内の設定温度と室内機の空気温と湿度をパラメータとして、潜熱および顕熱処理量テーブルを作成し、テーブル間については線形補完することとした。

なお、今回の簡易プログラムでは計算前の時間について値を保持しないため、内部保水量については扱わないこととした。

3.2.5.1 簡易計算モデルの作成

作成した簡易計算モデルについて説明する。

なお、エアコンが OFF の場合は、エアコンでの負荷処理は行われないため、ここでは、エアコンが ON 判定となっている場合についての、処理熱量等計算方法について説明する。

エアコンの処理では、以下の条件を必要とする。

(1) エアコン自体の持つ特性

$q_{c,max}$: エアコンの冷房最大能力[kW]

BF : バイパスファクター— (無単位)。0.15 とする。

また、表 56 のような表に対応する形で

$\bar{L}_{s,i,j}$: 顕熱処理量[W]

$\bar{L}_{l,i,j}$: 潜熱処理量[W]

を持つ。実際の値については、表 57 に示す値とした。

表 56 テーブルイメージ

吸い込み空気 温度(添字i)	23(j=0)		27(j=1)		31(j=2)		35(j=3)	
吸い込み空気 相対湿度 (添字j)	顕熱 処理量 [W]	潜熱 処理量 [W]	顕熱 処理量 [W]	潜熱 処理量 [W]	顕熱 処理量 [W]	潜熱 処理量 [W]	顕熱 処理量 [W]	潜熱 処理量 [W]
0.45(i=0)	$L_s[0,0]$	$L_l[0,0]$	$L_s[0,1]$	$L_l[0,1]$
0.55(i=1)	$L_s[1,0]$	$L_l[1,0]$
0.65(i=2)
0.75(i=3)	$L_s[3,3]$	$L_l[3,3]$

ここで添字iは吸い込み空気の相対湿度の水準 \bar{h}_{RM} 、添字jは吸い込み空気の温度の水準 $\bar{\theta}_{RM}$ を表し、

$$\bar{h}_{RM,i} = (0.45, 0.55, 0.65, 0.75) \quad (i = 0 \sim 3)$$

$$\bar{\theta}_{RM,j} = (23, 27, 31, 35) \quad (j = 0 \sim 3)$$

とする。

表 57 潜顕熱処理量テーブル [kW]

温度→ 湿度↓	24		27		30		33	
	顕熱	潜熱	潜熱	顕熱	潜熱	顕熱	潜熱	顕熱
50	0.300	0.000	1.011	0.000	1.943	1.357	1.864	1.436
60	0.300	0.000	0.848	0.000	1.658	1.642	1.569	1.731
70	0.300	0.000	0.688	0.154	1.399	1.901	1.304	1.996
80	0.300	0.000	0.524	0.349	1.162	2.138	1.062	2.238

(2) 人体から（リモコン操作で）渡される値

ON/OFF（継続を含む）：以下の処理は ON の場合に行う。

θ_{DSG} ：設定温度[°C] 当面 27.0°Cとする。

V_{AC} ：吹出し風量[m³/s] 当面 12.8m³/min（768m³/h）とする。

(3) 顕熱・潜熱処理量の算出

パラメータとして

θ_{RM} ：吸込み空気温度（前時刻の室空気温度）[°C]

h_{RM} ：吸込み空気相対湿度（前時刻の室空気相対湿度）[%]

が渡された場合、実際の顕熱処理量 L_s 、潜熱処理量 L_l （単位[W]）は以下の手順で求める。

(I) $\bar{\theta}_{RM,0} \leq \theta_{RM} \leq \bar{\theta}_{RM,3}$ かつ $\bar{h}_{RM,0} \leq h_{RM} \leq \bar{h}_{RM,3}$ の場合

温度・湿度とも表の範囲内におさまる場合に当たる。

$\bar{\theta}_{RM,j} \leq \theta_{RM} \leq \bar{\theta}_{RM,j+1}$ となるjおよび $\bar{h}_{RM,i} \leq h_{RM} \leq \bar{h}_{RM,i+1}$ となるiを求めておき、

$$L_{s,low} = (L_{s,i,j+1} - L_{s,i,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{s,i,j}$$

式 28

$$L_{s,high} = (L_{s,i+1,j+1} - L_{s,i+1,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{s,i+1,j}$$

式 29

$$L_s = (L_{s,high} - L_{s,low}) \times \frac{(h_{RM} - \bar{h}_{RM,i})}{0.1} + L_{s,low}$$

式 30

$$L_{l,low} = (L_{l,i,j+1} - L_{l,i,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{l,i,j}$$

式 31

$$L_{l,high} = (L_{l,i+1,j+1} - L_{l,i+1,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{l,i+1,j}$$

式 32

$$L_l = (L_{l,high} - L_{l,low}) \times \frac{(h_{RM} - \bar{h}_{RM,i})}{0.1} + L_{l,low}$$

式 33

として求める。

(II) $\bar{\theta}_{RM,0} \leq \theta_{RM} \leq \bar{\theta}_{RM,3}$ かつ $\bar{h}_{RM,3} \leq h_{RM}$ の場合

吸込み相対湿度が表の定義範囲を高い方に超過している場合にあたる。

$$L_s = (L_{s,3,j+1} - L_{s,3,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{s,3,j}$$

式 34

$$L_l = (L_{l,3,j+1} - L_{l,3,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{l,3,j}$$

式 35

(III) $\bar{\theta}_{RM,0} \leq \theta_{RM} \leq \bar{\theta}_{RM,3}$ かつ $h_{RM} \leq \bar{h}_{RM,0}$ の場合

吸込み相対湿度が表の定義範囲を低い方に超過している場合にあたる。

$$L_s = (L_{s,0,j+1} - L_{s,0,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{s,0,j}$$

式 36

$$L_l = (L_{l,0,j+1} - L_{l,0,j}) \times \frac{(\theta_{RM} - \bar{\theta}_{RM,j})}{4} + L_{l,0,j}$$

式 37

(IV) $\bar{\theta}_{RM,3} \leq \theta_{RM}$ かつ $\bar{h}_{RM,0} \leq h_{RM} \leq \bar{h}_{RM,3}$ の場合

吸込み空気温度が表の定義範囲を高い方に超過している場合にあたる。

$$L_s = (L_{s,i+1,3} - L_{s,i,3}) \times \frac{(h_{RM} - \bar{h}_{RM,i})}{0.1} + L_{s,i,3}$$

式 38

$$L_l = (L_{l,i+1,3} - L_{l,i,3}) \times \frac{(h_{RM} - \bar{h}_{RM,i})}{0.1} + L_{l,i,3}$$

式 39

(V) $\theta_{RM} \leq \bar{\theta}_{RM,0}$ かつ $\bar{h}_{RM,0} \leq h_{RM} \leq \bar{h}_{RM,3}$ の場合

吸込み空気温度が表の定義範囲を低い方に超過している場合にあたる。

$$L_s = (L_{s,i+1,0} - L_{s,i,0}) \times \frac{(h_{RM} - \bar{h}_{RM,i})}{0.1} + L_{s,i,0}$$

式 40

$$L_l = (L_{l,i+1,0} - L_{l,i,0}) \times \frac{(h_{RM} - \bar{h}_{RM,i})}{0.1} + L_{l,i,0}$$

式 41

(VI) $\bar{\theta}_{RM,3} \leq \theta_{RM}$ かつ $\bar{h}_{RM,3} \leq h_{RM}$ の場合

温度・湿度とも高い方に超過している場合にあたる。

$$L_s = L_{s,3,3}$$

式 42

$$L_l = L_{l,3,3}$$

式 43

(VII) $\bar{\theta}_{RM,3} \leq \theta_{RM}$ かつ $h_{RM} \leq \bar{h}_{RM,0}$ の場合

温度が高い方に、湿度が低い方に超過している場合にあたる。

$$L_s = L_{s,0,3}$$

式 44

$$L_l = L_{l,0,3}$$

式 45

(VIII) $\theta_{RM} \leq \bar{\theta}_{RM,0}$ かつ $\bar{h}_{RM,3} \leq h_{RM}$ の場合

温度が低い方に、湿度が高い方に超過している場合にあたる。

$$L_s = L_{s,3,0}$$

式 46

$$L_l = L_{l,3,0}$$

式 47

(IX) $\theta_{RM} \leq \bar{\theta}_{RM,0}$ かつ $h_{RM} \leq \bar{h}_{RM,0}$ の場合

温度・湿度とも低い方に超過している場合にあたる。

$$L_s = L_{s,0,0}$$

式 48

$$L_l = L_{l,0,0}$$

式 49

(4) 消費電力量の計算

パラメータとして

L_s : 顕熱処理量[W]

L_l : 潜熱処理量[W]

θ_{RM} : 吸込み空気温度 (前時刻の室空気温度) [°C]

h_{RM} : 吸込み空気相対湿度 (前時刻の室空気相対湿度) [%]

θ_0 : 外気温[°C]

を用いる。

(5) 吹出し温湿度の算出

吹出し空気温度 θ_{OUT} [°C]、吹出し空気絶対湿度 X_{OUT} [kg/kg]を次式で求める。

$$\theta_{OUT} = \theta_{RM} - \frac{L_s}{c_{air} \cdot \rho_{air} \cdot V_{AC}}$$

式 50

$$X_{OUT} = X_{RM} - \frac{L_l}{r_L \cdot \rho_{air} \cdot V_{AC}}$$

式 51

c_{air} : 空気比熱[J/kgK]。定数とする (1006)。

r_L : 水の蒸発熱[J/kg]。定数とする (2460×10³)。

ρ_{air} : 空気密度[kg/m³]。定数とする (1.2)。

X_{RM} : 吸込み空気絶対湿度 (前時刻の室空気相対湿度) [kg/kg]

X_{RM} は、 θ_{RM} と h_{RM} から以下の手順で求める。

- ① 温度 θ_{RM} [°C]の空気の飽和水蒸気圧 $P_{s, RM}$ [Pa]を求める (温度→飽和水蒸気圧の計算式参照)。
- ② 飽和水蒸気圧 $P_{s, RM}$ に対応する飽和絶対湿度 $X_{s, RM}$ [kg/kg]を求める (水蒸気圧→絶対湿度の計算式参照)。
- ③ $X_{RM} = X_{s, RM} \times h_{RM}$ とする。

(6) 蒸発温度の算出

エアコンの蒸発温度 θ_{evp} [°C]を以下の関係から求める。

$$\theta_{evp} \cdot (1 - BF) + \theta_{RM} \cdot BF = \theta_{OUT}$$

式 52

$$\therefore \theta_{evp} = \frac{\theta_{OUT} - \theta_{RM} \cdot BF}{1 - BF}$$

(7) 凝縮温度の算出

ある時点におけるエアコンの凝縮温度 θ_{cnd} [°C]は外気温に等しいとする。

$$\theta_{cnd} = \theta_o$$

式 53

なお、従来算出していたヒートポンプサイクルの過熱度・過冷却度は0とする。

(8) 理論効率の算出

エアコンの理論効率 e_{AC} を次式で求める。

$$e_{AC} = \frac{h_{evp,out} - h_{evp,in}}{h_{cmp,out} - h_{cmp,in}}$$

式 54

分母は圧縮機前後のエンタルピーの変化量、分子は蒸発器前後のエンタルピーの変化量である。
各変数は以下のように求める。

(I) $h_{cmp,in}$ の算出

まず、圧縮機入口温度 $\theta_{cmp,in}$ [°C]は蒸発温度に等しいとする。

$$\theta_{cmp,in} = \theta_{evp}$$

式 55

次に、蒸発圧力 P_{evp} [MPa]を以下で求める。

$$P_{evp} = f_{P,sgas}(\theta_{evp})$$

式 56

関数 $f_{P,sgas}(\theta)$ は後述する。

圧縮機入口圧力 $P_{cmp,in}$ [MPa]は P_{evp} に等しいとする。

$$P_{cmp,in} = P_{evp}$$

式 57

以上の値を用いて、圧縮機入口エンタルピー $h_{cmp,in}$ [kJ/kg]を次式で求める。

$$\therefore h_{cmp,in} = f_{cmp,in}(P_{cmp,in}, \theta_{cmp,in})$$

式 58

関数 $f_{cmp,in}(P, \theta)$ は後述する。

(II) $h_{cmp,out}$ の算出

まず、圧縮機入口エントロピー $S_{cmp,in}$ [kJ/kgK]を次式で求める。

$$S_{cmp,in} = f_{S,sgas}(P_{cmp,in}, h_{cmp,in})$$

式 59

関数 $f_{S,sgas}(P, h)$ は後述する。

圧縮機出口エントロピー $S_{cmp,out}$ [kJ/kgK]は $S_{cmp,in}$ に等しいとする。

$$S_{cmp,out} = S_{cmp,in} \quad \text{式 60}$$

凝縮圧力 P_{cnd} [MPa]を以下で求める。

$$P_{cnd} = f_{p,sgas}(\theta_{cnd}) \quad \text{式 61}$$

θ_{cnd} は前述の凝縮温度である。関数 $f_{p,sgas}(\theta)$ は蒸発圧力 P_{evp} を求める関数と同じ関数である（後述）。

圧縮機出口圧力 $P_{cmp,out}$ [MPa]は P_{cnd} に等しいとする。

$$P_{cmp,out} = P_{cnd} \quad \text{式 62}$$

これらから、圧縮機出口エンタルピー $h_{cmp,out}$ [kJ/kg]を次式で求める。

$$\therefore h_{cmp,out} = f_{cmp,out}(P_{cmp,out}, S_{cmp,out}) \quad \text{式 63}$$

関数 $f_{cmp,out}(P, S)$ は後述する。

(III) $h_{evp,in}$ の算出

まず、凝縮器出口温度 $\theta_{cnd,out}$ [°C]は凝縮温度に等しいとする。

$$\theta_{cnd,out} = \theta_{cnd} \quad \text{式 64}$$

凝縮器出口エンタルピー $h_{cnd,out}$ [kJ/kg]を以下で求める。

$$h_{cnd,out} = f_{liq}(P_{cnd}, \theta_{cnd,out}) \quad \text{式 65}$$

P_{cnd} は前述の凝縮圧力である。関数 $f_{liq}(P, \theta)$ は後述する。

蒸発器入口エンタルピー $h_{evp,in}$ [kJ/kg]は、凝縮器出口エンタルピー $h_{cnd,out}$ に等しいとして求める。

$$\therefore h_{evp,in} = h_{cnd,out} \quad \text{式 66}$$

(IV) $h_{evp,out}$ の算出

蒸発器出口エンタルピー $h_{evp,out}$ [kJ/kg]は、圧縮機入口エンタルピー $h_{cmp,in}$ に等しいとして求める。

$$\therefore h_{evp,out} = h_{cmp,in}$$

式 67

なお、 $e_{AC} > 10$ となる場合は $e_{AC} = 10$ とする。

(9) 冷媒状態の計算

各関数は次式で定義する。

図 129 に冷凍サイクルのイメージを示す。

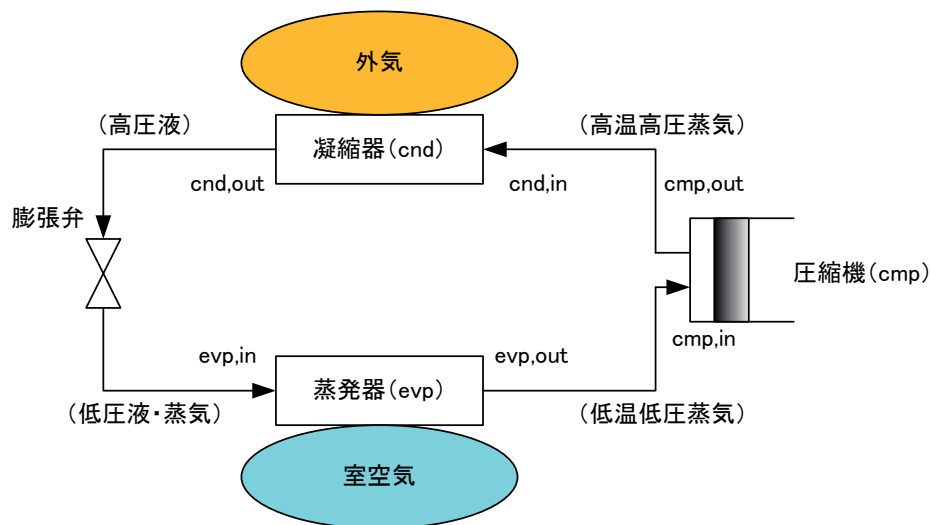


図 129 冷凍サイクルイメージ

(I) $f_{P,sgas}(\theta)$

飽和蒸気の温度 θ [°C]から圧力 [MPa]を求める。

$$\begin{aligned} f_{P,sgas}(\theta) = & 2.758\ 579\ 269\ 509\ 01 \times 10^{-17} \times \theta^8 \\ & + 1.493\ 820\ 579\ 117\ 53 \times 10^{-15} \times \theta^7 \\ & + 6.520\ 016\ 872\ 670\ 15 \times 10^{-14} \times \theta^6 \\ & + 9.141\ 530\ 349\ 999\ 75 \times 10^{-12} \times \theta^5 \\ & + 3.183\ 146\ 165\ 003\ 61 \times 10^{-9} \times \theta^4 \\ & + 1.607\ 035\ 666\ 630\ 19 \times 10^{-6} \times \theta^3 \\ & + 3.062\ 789\ 840\ 195\ 13 \times 10^{-4} \times \theta^2 \\ & + 2.544\ 619\ 929\ 920\ 37 \times 10^{-2} \times \theta^1 \\ & + 7.980\ 864\ 551\ 547\ 75 \times 10^{-1} \end{aligned}$$

式 68

(II) $f_{cmp,in}(P, \theta)$

圧縮機入口において蒸気の圧力 P [MPa]および温度 θ [°C]から比エンタルピー[kJ/kg]を求める。

$$\begin{aligned} f_{cmp,in}(P, \theta) = & -1.001\ 103\ 55 \times 10^{-1} \times P^3 \\ & -1.184\ 450\ 639 \times 10^1 \times P^2 \\ & -2.052\ 740\ 252 \times 10^2 \times P^1 \\ & +3.203\ 91 \times 10^{-6} \times T^3 \\ & -2.246\ 85 \times 10^{-3} \times T^2 \\ & +1.279\ 436\ 909 \times 10^0 \times T^1 \\ & +3.127\ 123\ 8 \times 10^{-2} \times P^2 \times T^1 \\ & -1.415\ 359 \times 10^{-3} \times P \times T^2 \\ & +1.055\ 539\ 12 \times 10^0 \times P^1 \times T^1 \\ & -1.949\ 505\ 039 \times 10^2 \end{aligned}$$

式 69

ただし、 $T = \theta + 273.15$ である。

(III) $f_{cmp,out}(P, S)$

圧縮機出口において蒸気の圧力 P [MPa]とエントロピー S [kJ/kgK]から比エンタルピー[kJ/kg]を求める。

$$\begin{aligned} f_{cmp,out}(P, S) = & -1.869\ 892\ 835\ 947\ 070 \times 10^{-1} \times P^4 \\ & +8.223\ 224\ 182\ 177\ 200 \times 10^{-1} \times P^3 \\ & +4.124\ 595\ 239\ 531\ 860 \times 10^0 \times P^2 \\ & -8.346\ 302\ 788\ 803\ 210 \times 10^1 \times P \\ & -1.016\ 388\ 214\ 044\ 490 \times 10^2 \times S^4 \\ & +8.652\ 428\ 629\ 143\ 880 \times 10^2 \times S^3 \\ & -2.574\ 830\ 800\ 631\ 310 \times 10^3 \times S^2 \\ & +3.462\ 049\ 327\ 009\ 730 \times 10^3 \times S \\ & +9.209\ 837\ 906\ 396\ 910 \times 10^{-1} \times P^3 \times S \\ & -5.163\ 305\ 566\ 700\ 450 \times 10^{-1} \times P^2 \times S^2 \\ & +4.076\ 727\ 767\ 130\ 210 \times 10^0 \times P \times S^3 \\ & -8.967\ 168\ 786\ 520\ 070 \times 10^0 \times P^2 \times S \\ & -2.062\ 021\ 416\ 757\ 910 \times 10^1 \times P \times S^2 \\ & +9.510\ 257\ 675\ 728\ 610 \times 10^1 \times P \times S \\ & -1.476\ 914\ 346\ 214\ 130 \times 10^3 \end{aligned}$$

式 70

(IV) $f_{s, gas}(P, h)$

蒸気の圧力 P [MPa]と比エンタルピー h [kJ/kg]からエントロピー[kJ/kgK]を求める。

$$\begin{aligned} f_{s, gas}(P, h) = & +5.823\ 109\ 493\ 752\ 840 \times 10^{-2} \times P^4 \\ & -3.309\ 666\ 523\ 931\ 270 \times 10^{-1} \times P^3 \\ & +7.700\ 179\ 914\ 440\ 890 \times 10^{-1} \times P^2 \\ & -1.311\ 726\ 004\ 718\ 660 \times 10^0 \times P \\ & +1.521\ 486\ 605\ 815\ 750 \times 10^{-9} \times h^4 \\ & -2.703\ 698\ 863\ 404\ 160 \times 10^{-6} \times h^3 \\ & +1.793\ 443\ 775\ 071\ 770 \times 10^{-3} \times h^2 \\ & -5.227\ 303\ 746\ 767\ 450 \times 10^{-1} \times h \\ & +1.100\ 368\ 875\ 131\ 490 \times 10^{-4} \times P^3 \times h \\ & +5.076\ 769\ 807\ 083\ 600 \times 10^{-7} \times P^2 \times h^2 \\ & +1.202\ 580\ 329\ 499\ 520 \times 10^{-8} \times P \times h^3 \\ & -7.278\ 049\ 214\ 744\ 230 \times 10^{-4} \times P^2 \times h \\ & -1.449\ 198\ 550\ 965\ 620 \times 10^{-5} \times P \times h^2 \\ & +5.716\ 086\ 851\ 760\ 640 \times 10^{-3} \times P \times h \\ & +5.818\ 448\ 621\ 582\ 900 \times 10^1 \end{aligned}$$

式 71

(V) $f_{liq}(P, \theta)$

過冷却液の圧力 P [MPa]と温度 θ [°C]から比エンタルピー[kJ/kg]を求める。

$$\begin{aligned} f_{liq}(P, \theta) = & +1.790\ 291\ 5 \times 10^{-2} \times P^3 \\ & +7.968\ 303\ 22 \times 10^{-1} \times P^2 \\ & +5.985\ 874\ 958 \times 10^1 \times P \\ & +9.866\ 77 \times 10^{-4} \times T^4 \\ & +9.805\ 167\ 7 \times 10^{-1} \times T \\ & -3.586\ 45 \times 10^{-3} \times P^2 \times T \\ & +8.231\ 22 \times 10^{-4} \times P \times T^2 \\ & -4.426\ 391\ 15 \times 10^{-1} \times P \times T \\ & -1.415\ 490\ 404 \times 10^2 \end{aligned}$$

式 72

ただし、 $T = \theta + 273.15$ である。

(10) 消費電力の算出

ある時点におけるエアコンの消費電力 E_{AC} [kW]は以下の手順で求める。

(I) 圧縮機の消費電力 E_{cmp} [kW]の算出

まず、圧縮機の圧縮効率 η を次式で求める。

$$\eta = -0.1255 \times (q_c)^2 + 0.6902 \times q_c + 0.0104$$

式 73

なお、本式の係数は実験結果の重回帰より求めた。データを図 130 圧縮機効率示す。

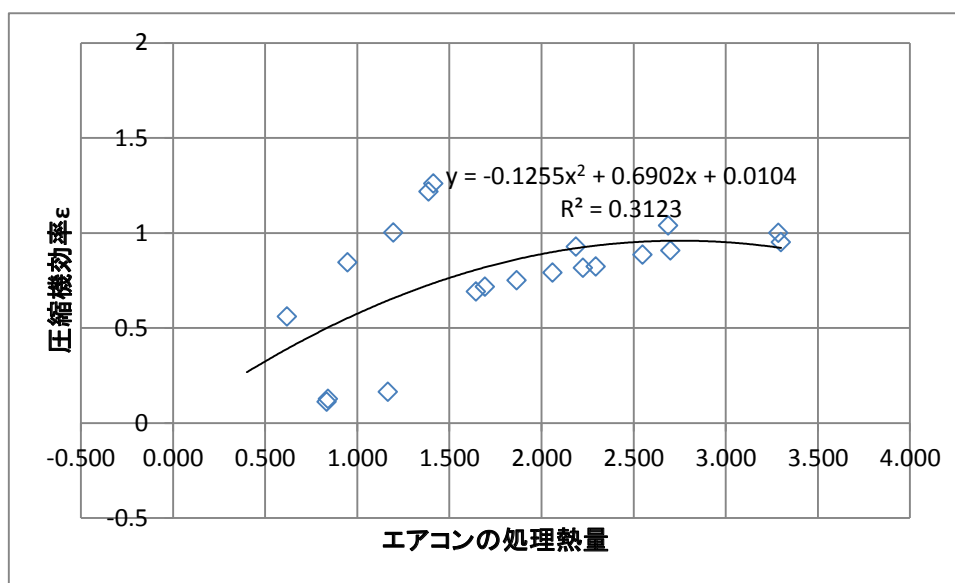


図 130 圧縮機効率

これと前項で求めた理論効率 e_{AC} から、次式で圧縮機の消費電力 E_{cmp} を求める。

$$E_{cmp} = \frac{q_c}{\eta \cdot e_{AC}}$$

式 74

なお、デフロスト係数および熱源機内の放熱損失項は省略した。

(II) 補機の消費電力 E_{aux} [kW]の算出

補機の消費電力は次式で求める。

$$E_{aux} = 0.0451$$

式 75

これらより、エアコンの消費電力 E [kW]は次式となる。

$$E_{AC} = E_{cmp} + E_{aux}$$

式 76

なお、消費電力を積算する際は、 E_{AC} に $(\Delta t/3600)$ を乗じて単位を[kWh]としておく。

3.2.5.2 簡易計算モデルの検証

計算プログラム用の簡易モデルにおける計算結果と実測結果を図 131 から図 133 に示す。よく一致しているといえる。

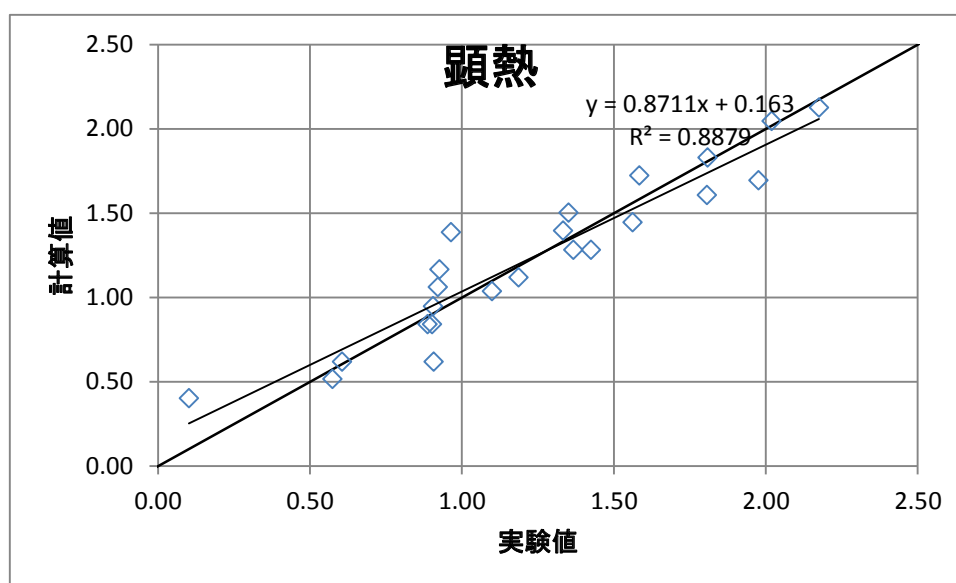


図 131 顕熱処理量の比較 (簡易モデル)

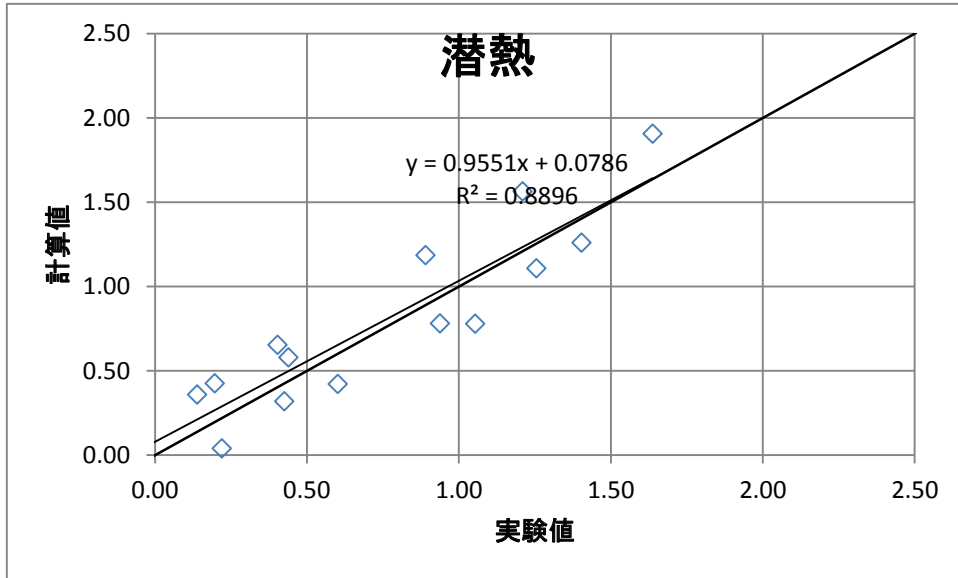


図 132 潜熱処理量の比較 (簡易モデル)

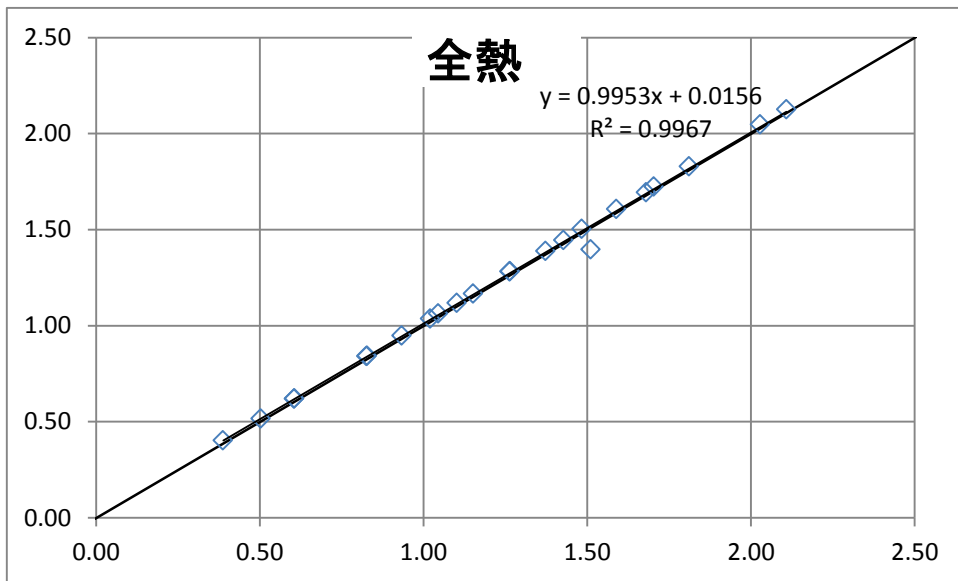


図 133 全熱処理量の比較 (簡易モデル)

3.2.6 まとめと課題

本研究で明らかになったことを以下に示す。

人工環境実験室実験より、ルームエアコンディショナーの除湿特性を明らかにするとともに、室内機熱交換器の状態を確認した。

また、確認した除湿特性と熱交換器状態などから、熱交換器の温度分布やルームエアコン室内機内部の保水量を考慮したモデルを作成し、実験に用いた機種において実測値と計算値が良く一致することを確認した。

さらに、作成したモデルによる計算結果から、室内機吸込み空気の時湿度と室内の冷房目標到達温度と顕潜熱量の関係を明らかにし、関係をテーブルとして持ち線形補完することで、計算速度の向上を図った簡易計算モデルを構築し、実験結果を再現することを確認した。

その他、実験中に当初は検討対象としていなかった、内部保水量について検討を進めることで、内部保水量の確認実験方法を構築した。その結果、室内機内部の保水量がエアコンの設置されている空間に対して影響を与えるに十分な量に達することを明らかにした。

一方で本結果の取り扱いについては、以下のような注意が必要である。

まず、ルームエアコンディショナーは日本国内だけでも多くのメーカーより、多数のラインナップが市場に出回っておりさらに、同じラインナップにおいても居住空間の広さ等に対応するために、機器定格能力が複数存在するが、モデルによる計算結果が検証できているのはあくまで、今回実験した機種のみであり、市場に存在するすべての機種について同様のモデルや同じパラメータが利用できるかについては未確認である。よって、省エネ法の計算モデルとするなど、機器を一般化する際は機器差について追加検証が必要である点に注意が必要である。

また、内部保水量については、機器差が大きいことや、水の供給源はあくまで潜熱負荷であることから、どのような状態でも今回確認されたような多量の内部保水量を有するわけではない点に注意が必要である。ただし、再放湿については課題が確認されたところであり、住宅特性や外部環境、内部発湿等の影響が大きいと考えられることから、評価に導入する前にさらなる確認が必要である。

なお、冷媒の冷凍サイクル中の圧力損失についても同様である。

以下については今回の研究では検討が不十分でありさらなる確認や検討が必要である。

今回実験に用いた機器においては、バイパスファクターなど同一の値を用いることができたが、この理由として、メーカーや定格性能は異なるが、市場に存在する機器のうち壁掛けタイプかつ、寸法規定のある機種であったため機器サイズが同様であったことなどのほかに、ラインナップのうち省エネ性を重視し選定したことから、上位機種であったことが影響している可能性は否定できない。今後の一般化においては、エネルギー消費効率の低い側の機器を確認する必要がある。この際今回は種々の影響を包括し多値として用いたバイパスファクターについては再度分解して検討する必要がある。

また、室内機風量の問題がある。これは、機器の設計思想等による制御の違いもあるが、除湿による室内機熱交換器への結露により風量が低下することが分かった。さらなる精緻化では、風量が機器省エネルギー性能に与える影響は大きいことから、制御によらない風量の取り扱いについてはさらなる検討が必要といえる。

3.3 ビルマルチエアコンの潜熱処理特性の把握

3.3.1 潜熱処理特性を含むビルマルチエアコン特性式

従来のビルマルチエアコンでは、冷房時の蒸発温度は固定制御を行っており、室内機吸込み空気温湿度条件が決まると、室内機のSHF（Sensible Heat Factor：顕熱比）は概ね一定であった。しかし蒸発温度を変化させた場合、室内機のSHF（およびLHF：Latent Heat Factor）は大きく変動する（図 134 参照）。ここでは、より難易度の高い蒸発温度可変制御を導入したビルマルチエアコンの計算モデルを構築することで、潜熱処理量の変動が大きい場合にも適用可能なビルマルチエアコンの計算モデルを作成することで、より汎用的に潜熱処理量を予測可能なビルマルチエアコンの特性式を提案する。

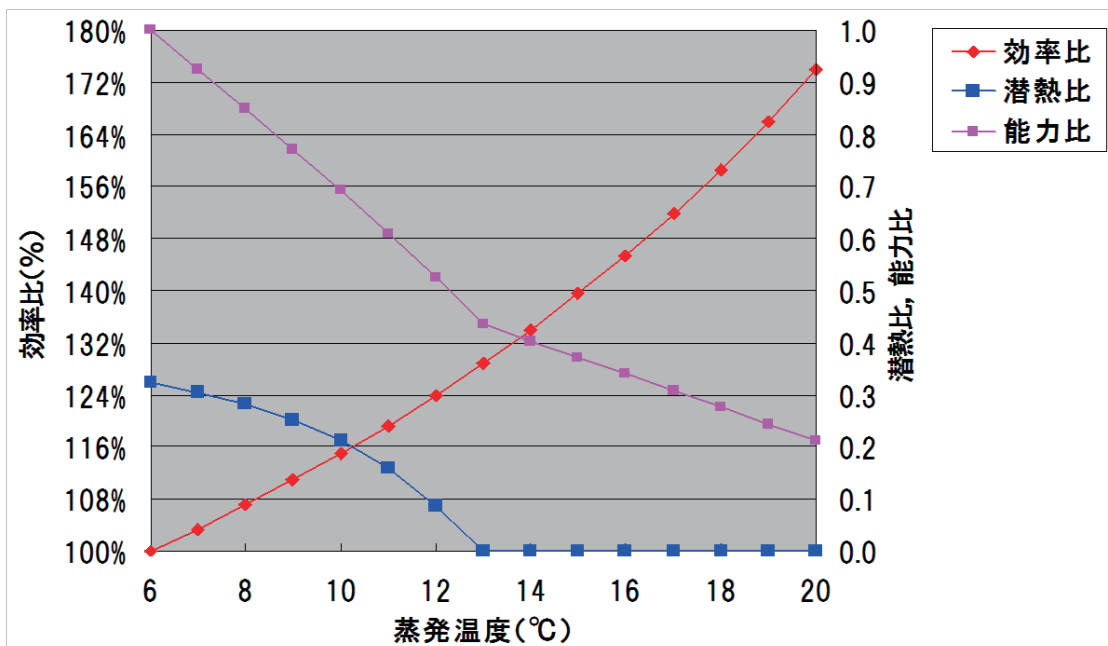


図 134 蒸発温度と潜熱比 (LHF)・能力・効率の関係

3.3.1.1 冷媒温度可変方式のビルマルチエアコンの性能計算モデル検討

蒸発温度固定制御方式のビルマルチエアコンについては、既存の計算モデルが存在するが蒸発温度可変制御方式のビルマルチエアコンに同じ計算モデルを用いた場合、図 135 に模式的に示したように最も負荷が高いゾーンに設置された室内機の能力がゾーンで発生する負荷を処理できる冷媒温度を決定する。即ち、最も負荷率が大きい室内機に合わせた蒸発温度で空調システム全体が運転されるため、ゾーン毎に分割した計算モデルが必要となる。

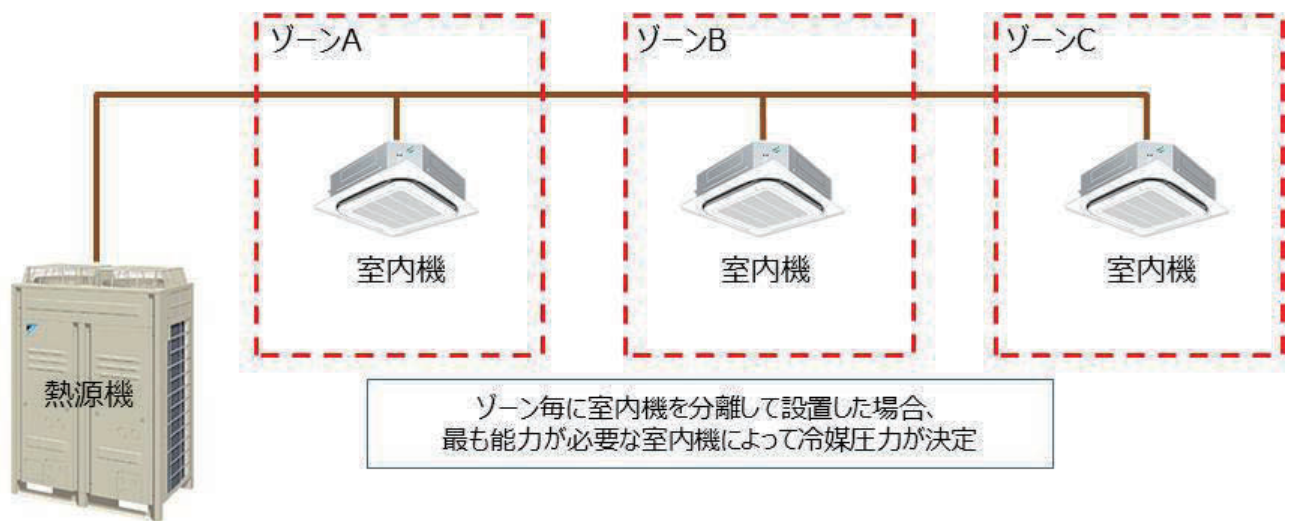


図 135 冷媒温度可変制御方式模式図

そこで、今回新たに室内機－配管－室外機に分割した計算モデルを作成した。

当該計算モデルによる、冷房運転時の計算構成を図 136 に示す。図に示すように、室内機、枝配管、主配管、室外機にシステムを分割して計算を行う。必要負荷に併せて各室内機の冷媒蒸発温度を仮決定した後に配管合流部で最も蒸発温度が低圧となる室内機を選定するとともに、各室内機ファン風量および冷媒循環量を算出し、各室内機で必要とされる冷媒循環量の合計をシステム全体の冷媒循環量とする。

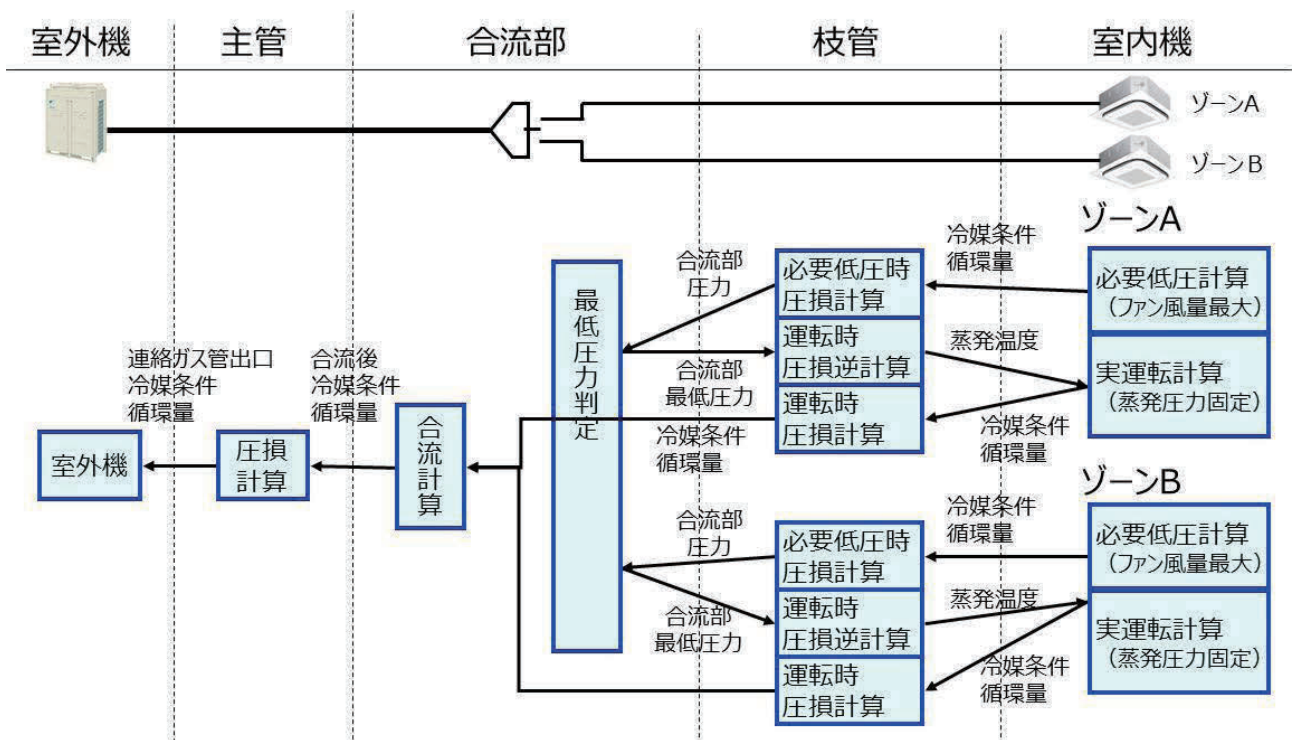


図 136 ビルマルチエアコン計算モデル

3.3.1.2 室内機計算モデル

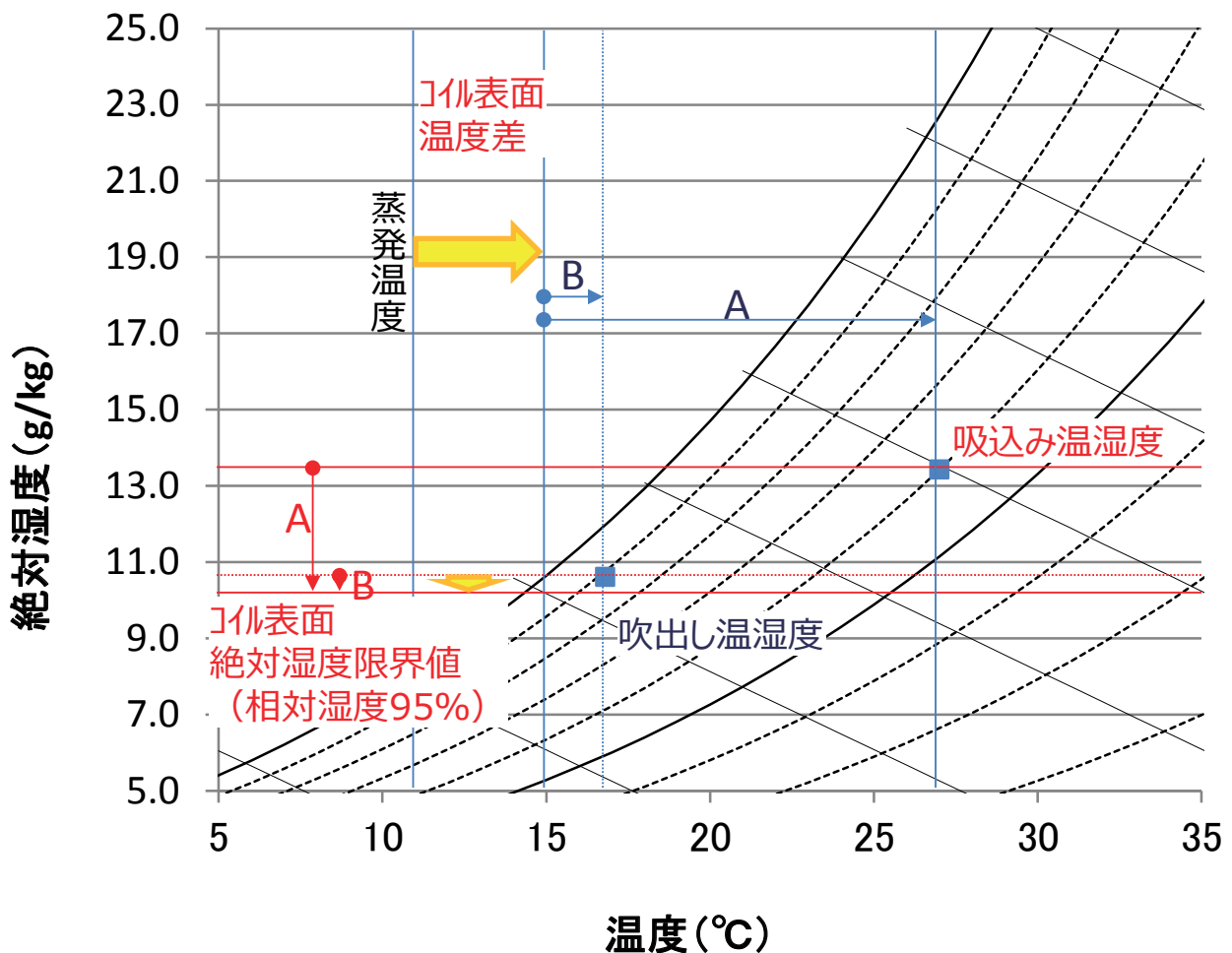
空調機の潜熱処理（除湿）を行うのは冷房時の室内熱交表面における結露であるため、潜熱処理を

モデル化するためには、室内熱交換器のモデルが重要な役割を果たす。

3.2 のエアコンのモデル化に際しては、フィンコイル熱交換器の冷媒流に沿って、伝熱管を微小区間に分割して計算する方法を採用した。エアコンの室内機はほとんどが壁掛け型であり、多少計算時間が長くなっても、壁掛け型のラインナップに合わせて機器性能計算を実施し、性能特性を表にすることでシミュレーションに組み込むことが出来る。しかしビルマルチ空調機に接続される室内機は種類・容量ともに多岐にわたる為、詳細計算を元にしたモデルで計算を行うのは現実的ではない。そこで、ビルマルチ用の室内機モデルとして、既存の国交省LCEMツールなどに用いられているバイパスファクターモデルをベースに検討を行った。

(1) 既存のバイパスファクターモデル

ビルマルチエアコンの技術資料に記載されているバイパスファクターを用いて、室内機の吹出し空気温湿度を予測し、室内機吸込み空気温湿度と吹出し空気温湿度のエンタルピー差と風量から室内機の能力を算出している。図 137 に空気線図上の動きを示す。



3.2.3.3 項のエアコンの熱交換器モデルでは、熱交換器を多数の微小領域に分割して計算しているため、各微小領域の計算中には空気物性の変化を無視できるものと考え、比熱などの値を用いた計算を行っているが、当ビルマルチエアコンのモデルでは熱交換器を分割することなく評価しようとしているため、比熱などは用いずに熱交換器入口・出口の空気のエンタルピーから、潜熱能力・顕熱能力を算出している。

図に示すように、入力値として

- ・ 空調機に流通する冷媒の蒸発温度
- ・ コイル表面と冷媒蒸発温度の温度差
- ・ バイパスファクター

を与える。

AH (T,RH)	: 絶対湿度	[g/kg]
Te	: 冷媒蒸発温度	[°C]
Tc	: 冷媒凝縮温度	[°C]
ΔT	: コイル表面温度差	[°C]
Hlim	: コイル表面絶対湿度限界値	[kg/kg] : Te + ΔT における相対湿度 95%の絶対湿度
BPF	: バイパスファクター	
Tin	: 吸込み温度	[°C]
Hin	: 吸込み湿度	[g/kg]
Tout	: 吹出し温度	[°C]
Hout	: 吹出し湿度	[g/kg]

<冷房時>

$$T_{out} = T_{in} - (T_{in} - (T_e + \Delta T)) \times (1 - BPF) \quad \text{式 77}$$

$$H_{out} = H_{in} - (H_{in} - \text{MIN}(H_{lim}(T_e + \Delta T), H_{in})) \times (1 - BPF) \quad \text{式 78}$$

<暖房時>

$$T_{out} = T_{in} + ((T_c - \Delta T) - T_{in}) \times BPF \quad \text{式 79}$$

$$H_{out} = H_{in} \quad \text{式 80}$$

本調査では、実測データとシミュレーション結果を比較した結果として、コイル表面温度差を 4°C に設定している。

また、従来のビルマルチエアコンの冷媒蒸発温度は 6°C、高顕熱形ビルマルチエアコンの冷媒蒸発温度は 11°C に設定している。

(2) 室内機計算モデルの作成

上記既存の室内機モデルでは、コイル表面空気温度と冷媒温度（蒸発温度、凝縮温度）の差（以下 ΔT ）とバイパスファクター（以下 BPF）の二つの固定値から給気の状態を計算し、給気温度と室温の差から要求顕熱負荷を満たすように室内機風量を決定している。一方、 ΔT と BPF からの能力予測では、室内機の過熱度（以下 SH）、過冷却度（以下 SC）の変化による能力変化が考慮されていなかった。

そこで、

- ・ SH、SC の変化による能力変化を補正する方式の確立
- ・ ΔT と BPF の機種ごとの差分の分析とデータ作成のための試験項目の検討を行うために実測試験を行った。

(I) 試験装置

試験装置の概要を図 138 に示す。

本試験装置は JISB8616 の Air-Enthalpy (AE) 法に基づき、室内空気（吸込空気）と室内機熱交換器を通過した空気（吹出空気）の温度・湿度変化（顕熱・潜熱変化）と空気重量の積で室内機能力を求める。

室外温調室と室内温調室は任意の温度・湿度条件に設定可能であり、室内機・室外機の吸込空気の温度・湿度を一定に制御している。

室内機取付けは、室内機を受風チャンバーに接続して吹出空気を AE チャンバーに導くように設置し、室内機吹出空気の温湿度と体積風量を測定する。AE チャンバーでは、誘引ファンにより室内機の吹出圧力を設定した。（天井型室内機、壁掛機では 0Pa、ダクト機では 50Pa）

また受風チャンバーから AE チャンバーまでに生じる熱損については、予め能力校正機を使った能力測定で冷房・暖房毎に係数を決定し、室内空気温度と吹出空気温度の差との積で補正量を求めている。

室外機は風量測定等実施しないため解放状態で設置し、室内機と室外機の連絡配管は各試験とも長さ 20m の断熱配管を使用している。

また装置外には大気圧計があり、大気圧変化による空気密度の補正を行なう。

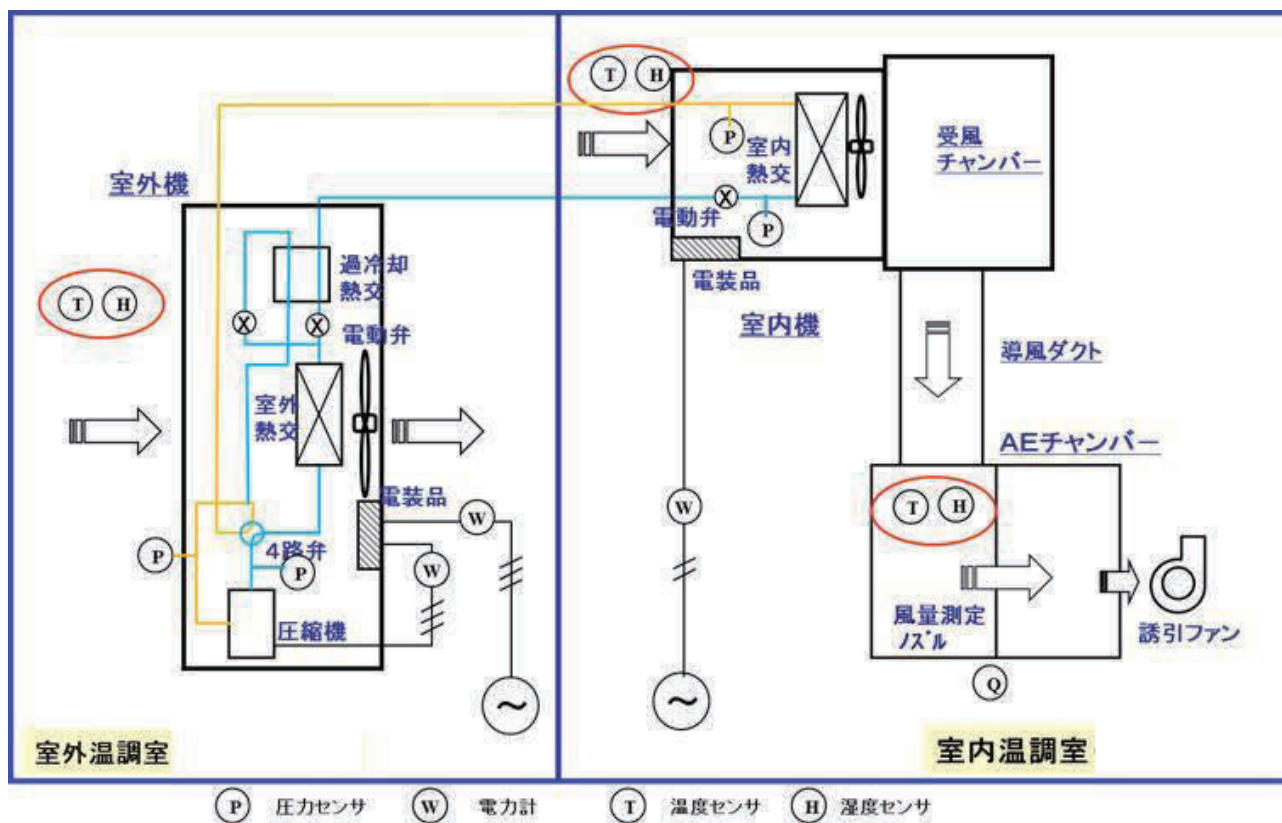


図 138 試験装置概要図

(II) 試験方法

空調機能力の測定は以下の手順で行う。本試験では冷媒蒸発温度（ T_e ）が室内空気露点近くまたはそれ以上になる条件での測定も行った。

- ① 室内機を受風チャンバーに取り付ける。
- ② 室外温調室に室外機を設置し、室内機と連絡配管で接続する。
- ③ 空調機システムを真空排気後、室外機初期充填量+液管長補正の冷媒を充填する。
- ④ 室内温調室、室外温調室の空気条件を目標値に設定する。
- ⑤ 空調機の運転を開始する。そのとき下記が目標値になるよう機器を制御する。

・ 室内機風量	(室内機ファン回転数)
・ 冷房時冷媒蒸発温度	(室内機熱交入口サーミスタ)
・ 熱交出口 SCH	(室内機熱交出口サーミスタ)
・ 室内機吹出抵抗	(AE チャンバー静圧)
・ 室外機圧縮機回転数	
・ 圧縮機吸入 SH	(吸入サーミスタ)
・ 室外機過冷却後冷媒温度	(冷房時のみ)
・ 圧縮機吐出圧	(凝縮温度、暖房時のみ)

- ⑥ 機器の安定運転を 20 分程度継続した後、室内機吹出温湿度が一定の状態を 15 分程度能力測定を行う。室内機能力はこの時間平均値とする。ただし、冷房条件で冷媒蒸発温度が室内空気露点と近い条件では熱交フィン表面及びドレンパンに残ったドレン水の再蒸発が潜熱能力に影響を与えるので安定運転の時間を 1 時間程度と長めにとるようにした。
- ⑦ ⑤の目標値設定を変更し、能力測定を繰り返す。

測定で得られたデータから室内機能力の計算は以下の手順で行う。

- ① 室内機吸込空気および吹出空気の温度・湿度から顕熱エンタルピー変化 ΔH_s と潜熱エンタルピー変化 ΔH_L を求める。
- ② 室内空気と吹出空気の温度差から能力補正值 Q_{ho} を求める。
- ③ AE チャンバーでは、ノズル前後差圧から体積流量 Q_v を、温度・湿度から密度 ρ を求める。それらの積に大気圧補正を加えて吹出空気の質量流量 Q_m を求める。
- ④ 空調機顕熱能力 Q を顕熱能力 Q_s と潜熱能力 Q_L の和で求める。計算式は下記のようになる。

$$Q_s = \Delta H_s \cdot Q_m + Q_{ho}$$

式 81

$$Q_L = \Delta H_L \cdot Q_m$$

式 82

$$Q = Q_s + Q_L$$

式 83

(III) 室内機冷房・暖房能力測定条件

特性式の係数は基本的には実機の能力測定から求める。

必要な能力測定条件を表 58 に示す。

冷房時は蒸発温度の範囲によって潜熱処理の有無が変わるため、それによる挙動の変化が予測される。そこで、 T_e 条件は定格（潜熱顕熱ともに処理）・主に顕熱を処理・顕熱のみ処理するそれぞれの範囲の値を選定した。逆に暖房時はその必要が無いため、 T_e は 2 条件としている。

また、SCH（熱交換器出口冷媒の過冷却度・過熱度）の影響は計測結果より二次式で表す必要性があったため、最小でも 3 条件での計測が必要となった。

表 58 特性式用能力測定条件

運転モード	室内空気条件 D.B./W.B	風量 タップ	T_{ec} 温度	SCH	特性	試験 番号
冷房 (6 点)	27/19	H	6	3	定格条件	①
		H	11	3	T_{ec} 特性把握	②
		H	20	3	T_{ec} 特性把握	③
		H	6	5	SCH 特性把握	④
		H	6	10	SCH 特性把握	⑤
		L	6	3	風量特性確認	⑥
暖房 (5 点)	20/13.5	H	46	5	定格条件	⑦
		H	38	5	T_{ec} 特性把握	⑧
		H	46	12	SCH 特性把握	⑨
		H	46	20	SCH 特性把握	⑩
		L	46	5	風量特性確認	⑪

これらの試験は AE 装置を備えた試験設備で行い、風量・吹出し温湿度も計測するものとする。AE 装置の機外静圧条件は室内機毎の定格条件に合わせた。

工数上全点の測定ができない場合、あるいは旧型機の特性式を作成するにあたっては、必要に応じてシミュレーションによる補外、補間を行う。また同一の熱交換器、筐体を有する室内機（称呼能力、風量違いの機種）では SCH 特性や風量特性も相似であり、精度上問題なければ係数を共有することも可能と考えられるが、試験結果の蓄積による検証が必要である。

今回試験を行った室内機、室外機の組合せを表 59 に示す。

表 59 供試機型式

	被試験機タイプ	室内機型式	室外機型式
①	天カセ（ラウンドフロー）	FXYFP71MC	RXYP140AA (5HP)
②		FXYFP56MC	
③	ダクト	FXYMP71A	
④	壁掛	FXYAP71M	
⑤	天カセ（ダブルフロー）	FXYCP71M	

(IV) 特性式の形式

能力特性式は室内機の顕熱能力および潜熱能力を室内機の状態変数を用いて算出する。また室内機の全熱能力は顕熱能力と潜熱能力の和に等しい。

なるべくエアネット・RAMモニタで取得できる変数を選定し、表 60 に示す状態変数を計算に用いる。

表 60 各種データの取得方法

状態変数	取得方法
吸込み空気温度[°C]	室内機センサー『TH 1』で計測
吸込み空気絶対湿度[g/kg]	これのみ既存センサーで計測不可 運転中の能力を予測するためには、 別途湿度センサーの設置などの検討が必要
冷媒飽和温度[°C]	室内機センサー『TH 2』で計測
SH/SC[°C]	室内機センサー『TH 3』と『TH 2』の差分
質量風量[kg/s]	ファン回転数から風量[m ³ /min]を、 吸込み空気温湿度から空気密度[kg/m ³]を算出して乗算

SH、SC の影響については、蒸発温度制御の導入により、SH、SC の制御目標値が運転条件によって変わる制御になっていることから、その分の熱交換能力補正項が必要と考え、実機計測の結果を元に SH/SC の値の式で表される補正値をコイル表面空気温度の算出時に加えるものとした。

Q	: 全熱能力	[kW]
Q _s	: 顕熱能力	[kW]
Q _l	: 潜熱能力	[kW]
ΔH	: 入口出口エンタルピー差	[kJ/kg]
ΔH _s	: 入口出口顕熱エンタルピー差	[kJ/kg]
ΔH _l	: 入口出口潜熱エンタルピー差	[kJ/kg]
G _{va}	: 質量風量	[kg/sec]
G _f	: 体積風量	[m ³ /sec]
ρ(T,AH)	: 空気密度	[kg/m ³]
f(T,AH)	: 空気エンタルピー	[kJ/kg]
T _{out}	: 吹出し温度	[°C]
T _{in}	: 吸込み温度	[°C]
T _{surf}	: コイル表面空気温度	[°C]
AH _{out}	: 吹出し絶対湿度	[g/kg]
AH _{in}	: 吸込み絶対湿度	[g/kg]
AH _{surf}	: コイル表面空気絶対湿度	[g/kg]
BPF	: バイパスファクター	
T _e	: 冷媒蒸発飽和温度	[°C]
T _c	: 冷媒凝縮飽和温度	[°C]
T _{const}	: コイル表面温度差基準値	[°C]
AH (T,RH)	: 絶対湿度	[g/kg]

$$Q = G_{va} \times \Delta H \quad \text{式 84}$$

$$Q_s = G_{va} \times \Delta H_s \quad \text{式 85}$$

$$Q_l = G_{va} \times \Delta H_l \quad \text{式 86}$$

$$G_{va} = G_f \times \rho (T_{in}, AH_{in}) \quad \text{式 87}$$

$$\Delta H = \Delta H_s + \Delta H_l \quad \text{式 88}$$

$$\Delta H_s = f(T_{out}, AH_{out}) - f(T_{in}, AH_{out}) \quad \text{式 89}$$

$$\Delta H_l = f(T_{in}, AH_{out}) - f(T_{in}, AH_{in}) \quad \text{式 90}$$

$$T_{out} = T_{in} \times BPF + T_{surf} \times (1 - BPF) \quad \text{式 91}$$

$$AH_{out} = AH_{in} \times BPF + AH_{surf} \times (1 - BPF) \quad \text{式 92}$$

【蒸発器】

$$T_{surf} = T_e + T_{const} + g(SH) \quad \text{式 93}$$

$$g(SH) = a \times SH^2 + b \times SH \quad \text{式 94}$$

$$AH_{surf} = \min (AH_{in}, AH (T_{surf}, 98)) \quad \text{式 95}$$

【凝縮器】

$$T_{surf} = T_c - T_{const} - g(SC) \quad \text{式 96}$$

$$g(SC) = a \times SC^2 + b \times SC \quad \text{式 97}$$

$$AH_{surf} = AH_{in} \quad \text{式 98}$$

ここで、

これらの内、室内機の特性を表すパラメータは、

- BPF : バイパスファクター
- T_{const} : コイル表面温度差基準値 [°C]
- a, b : SH, SC の影響を表す式の係数

であり、且つ試験結果に照らした結果、冷房と暖房は共用不可である。よって熱交特性を表すパラメータ数は、冷房 4、暖房 4 の計 8 個必要となる。

特性式の係数を表 61～表 65 に示す。

表 61 壁掛 7.1kW 係数

	コイル表面 相対湿度上限	バイパス ファクタ	コイル表面温度－冷媒温度 $\Delta T = a \times SCH^2 + b \times SCH + c$		
			a	b	c
冷房	98%	0.291	0	0.329	1.914
暖房	—	0.262	0	0.455	0

表 62 天埋ダクト 7.1kW

	コイル表面 相対湿度上限	バイパス ファクタ	コイル表面温度－冷媒温度 $\Delta T = a \times SCH^2 + b \times SCH + c$		
			a	b	c
冷房	98%	0.161	0	0.597	0
暖房	—	0.136	0	0.484	0

表 63 天埋カセットラウンドフロー型 5.6kW

	コイル表面 相対湿度上限	バイパス ファクタ	コイル表面温度－冷媒温度 $\Delta T = a \times SCH^2 + b \times SCH + c$		
			a	b	c
冷房	98%	0.188	0	0.478	0.026
暖房	—	0.160	0	0.326	0.231

表 64 天埋カセットラウンドフロー型 7.1kW

	コイル表面 相対湿度上限	バイパス ファクタ	コイル表面温度－冷媒温度 $\Delta T = a \times SCH^2 + b \times SCH + c$		
			a	b	c
冷房	98%	0.050	0.0853	0	0
暖房	—	0.131	0	0.213	0

表 65 天埋カセットラウンドフロー型 14.0kW

	コイル表面 相対湿度上限	バイパス ファクタ	コイル表面温度－冷媒温度 $\Delta T = a \times SCH^2 + b \times SCH + c$		
			a	b	c
冷房	98%	0.210	0.088	-0.769	5.769
暖房	—	0.072	0	0.487	0.072

(V) 実測値と計算モデルの比較

実測値と計算モデルの比較を図 139~図 147 に示す。

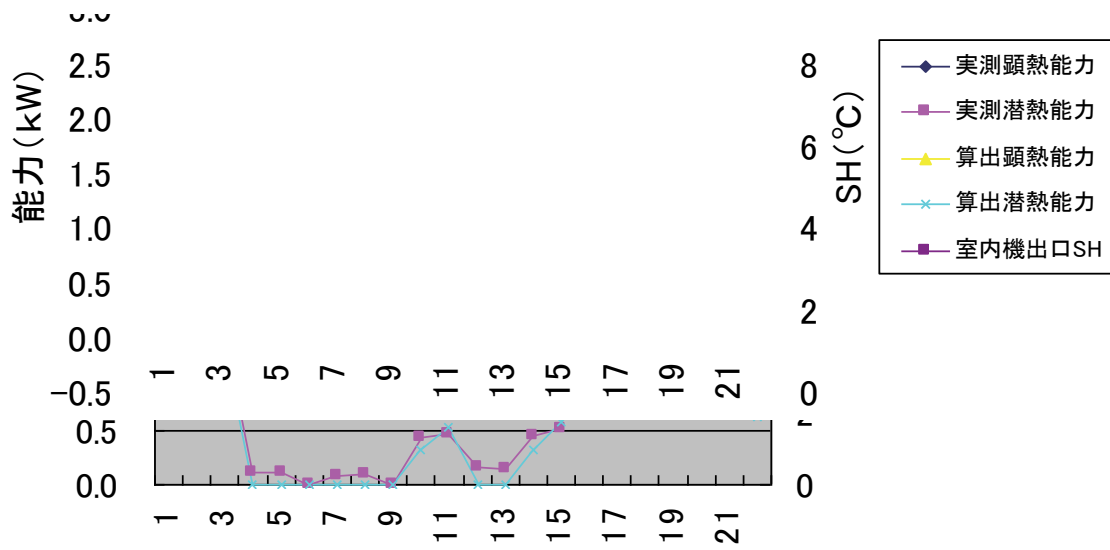


図 139 壁掛け 7.1kW (冷房運転)

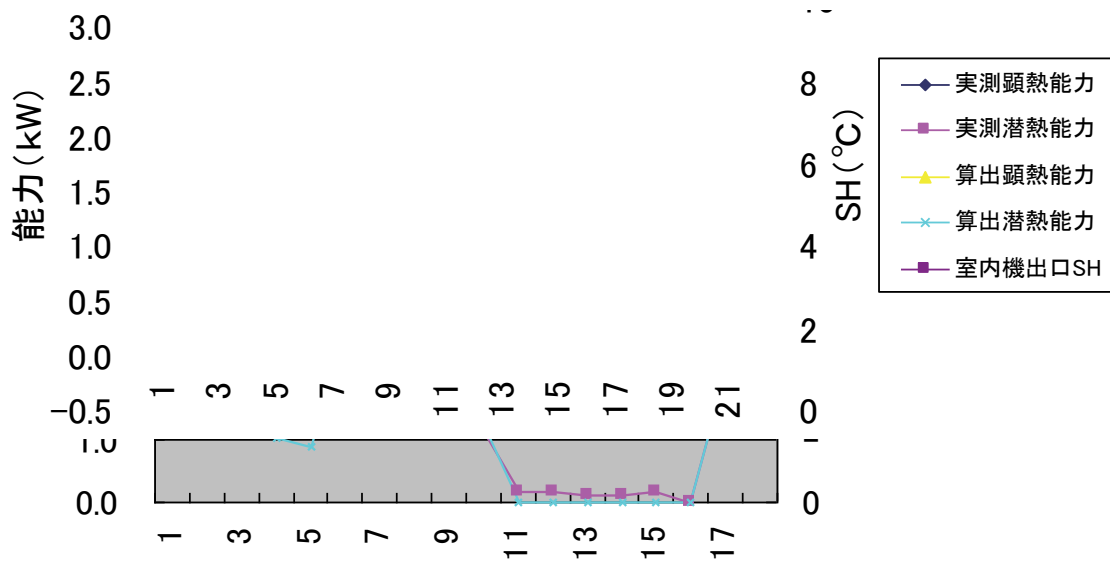


図 140 天埋ダクト 7.1kW (冷房運転)

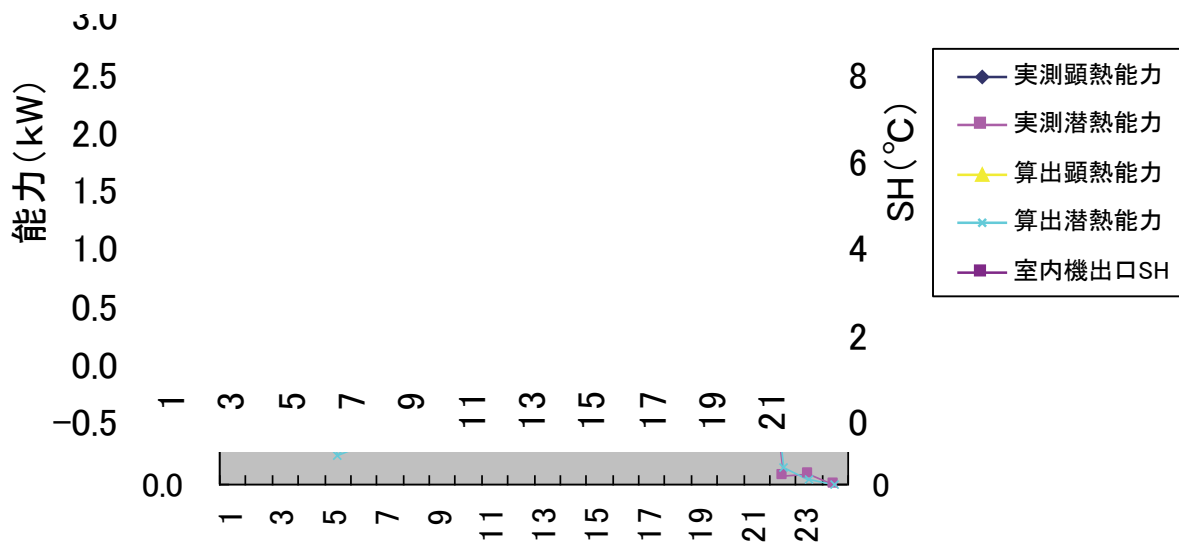


図 141 天埋カセットラウンドフロー型 5.6kW (冷房運転)

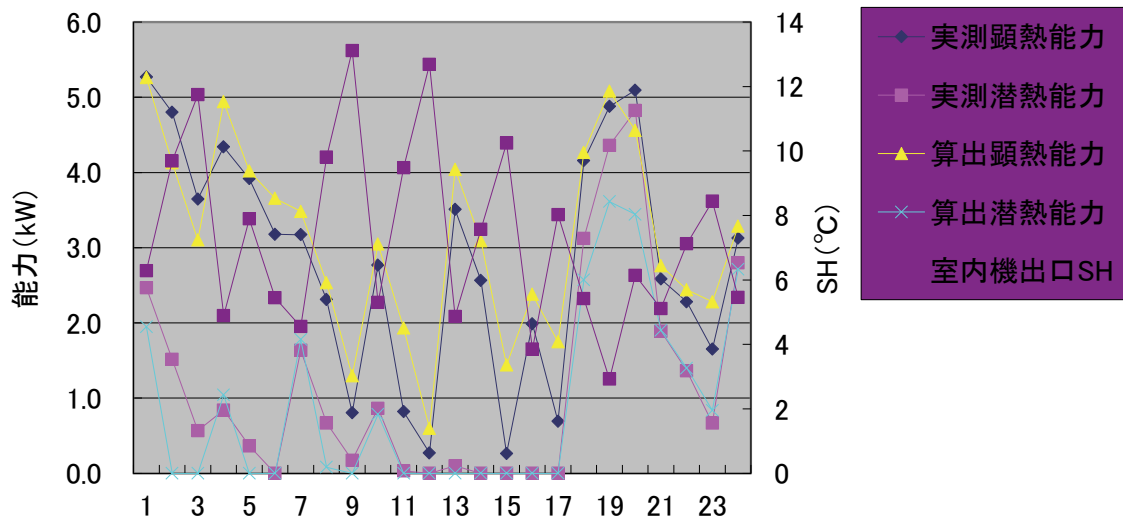


図 142 天埋カセットラウンドフロー型 7.1kW (冷房運転)

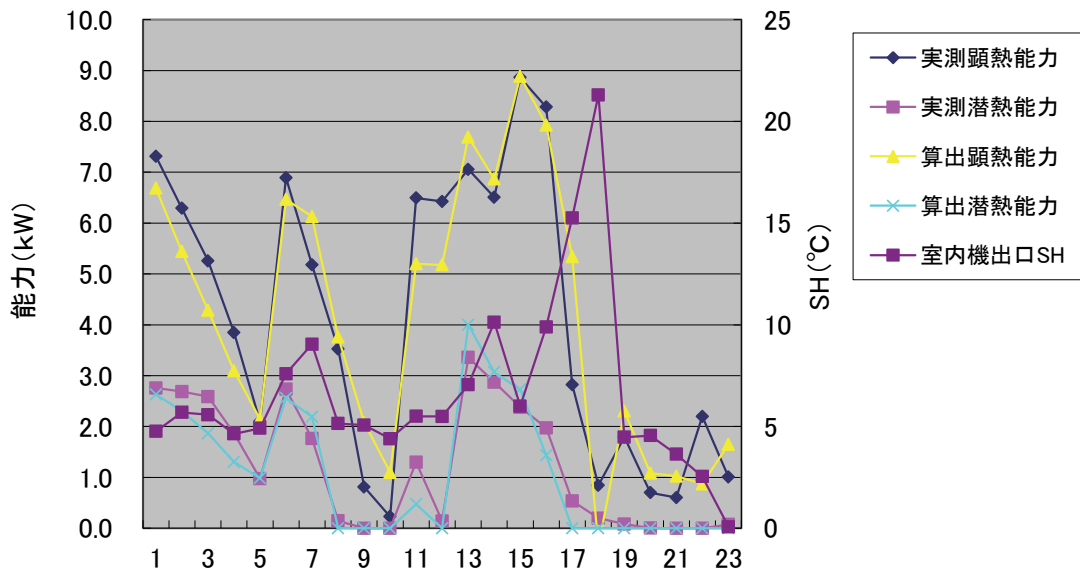


図 143 天埋カセットラウンドフロー型 14.0kW (冷房運転)

中・大型のラウンドフローで特に SH が大きい領域 (10°C以上) では乖離が見られるが、概ね計算モデルと実測結果は良好に合致している。

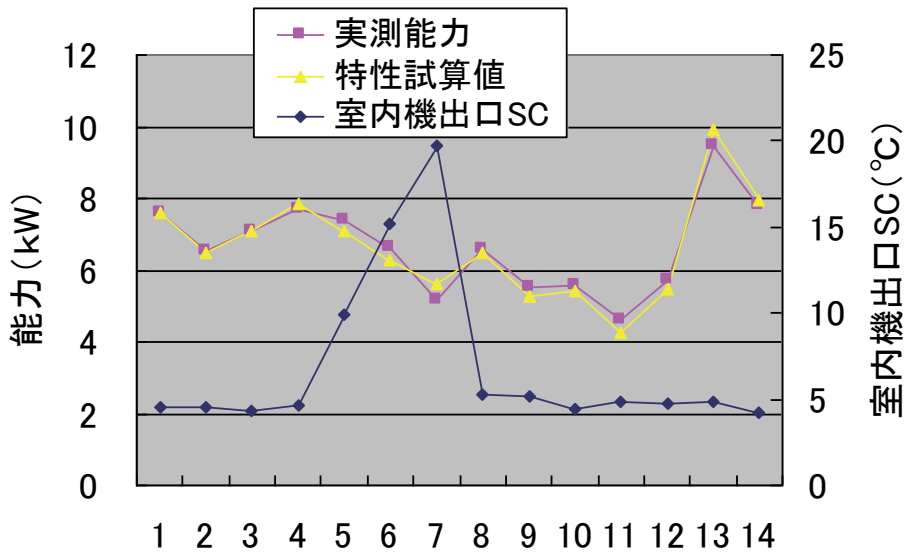


図 144 天埋ダクト 7.1kW (暖房運転)

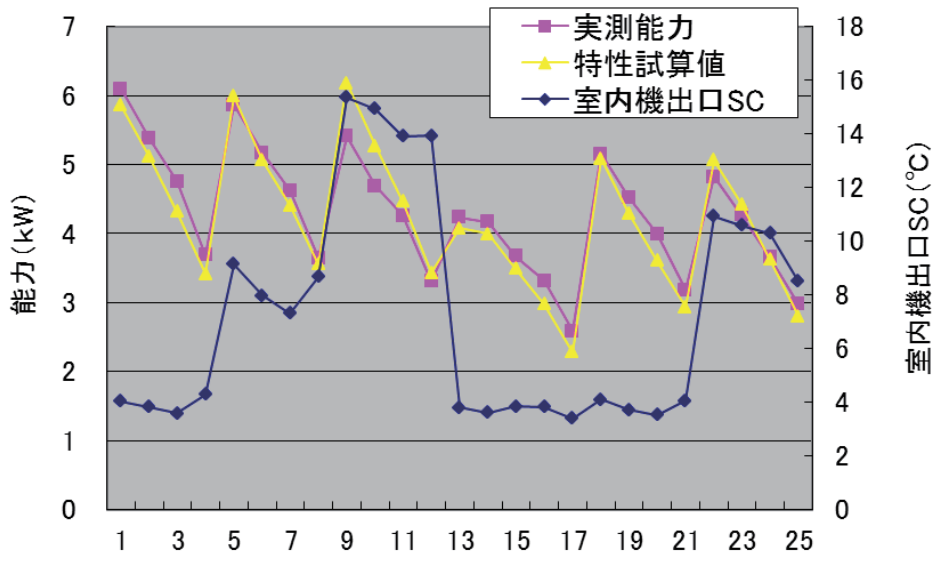


図 145 天埋カセットラウンドフロー型 5.6kW (暖房運転)

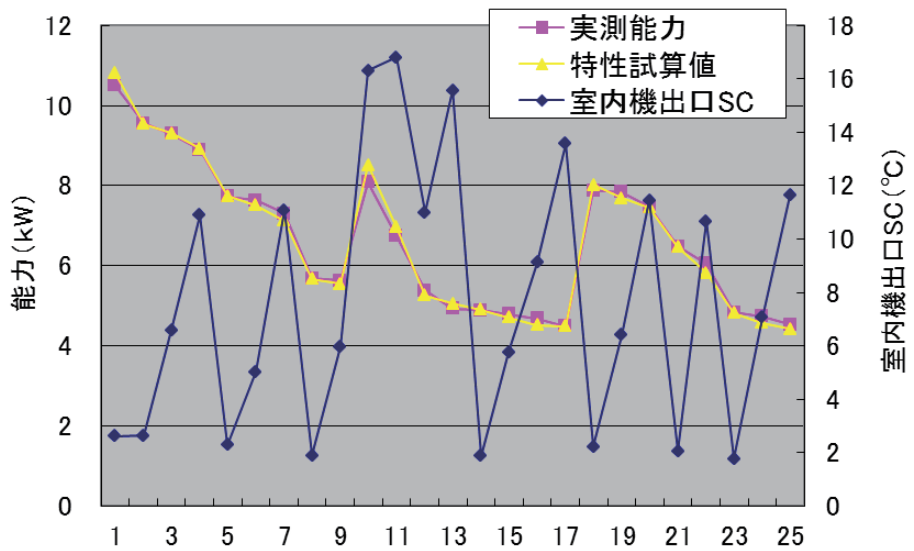


図 146 天埋カセットラウンドフロー型 7.1kW (暖房運転)

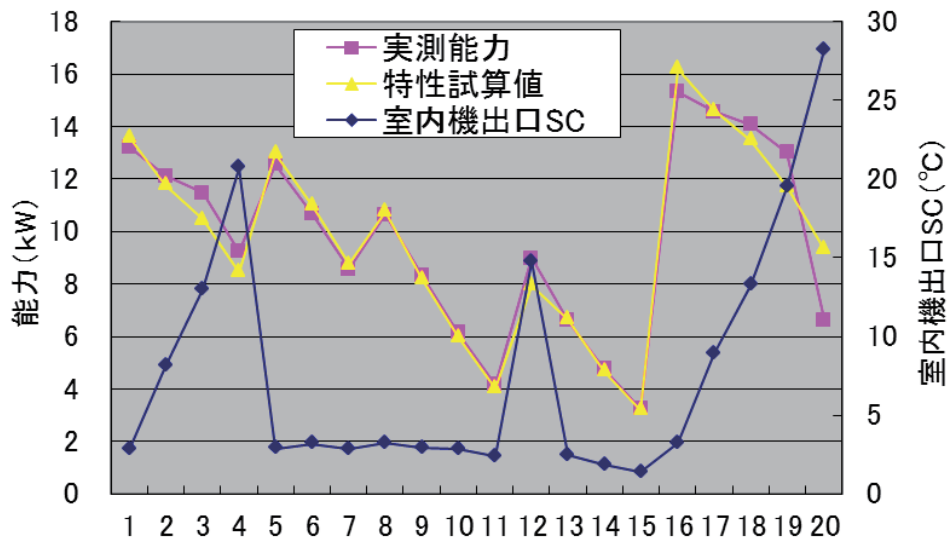


図 147 天埋カセットラウンドフロー型 14.0kW (暖房運転)

暖房運転においては、計算モデルと実測結果は良好に一致した。

3.3.1.3 連絡配管計算モデル

連絡配管の圧力損失・熱損失の計算には一般的な理論式を用い、以下のモデルを採用した。

Flux	: 質量流量	[kg/m ² /sec]
Gr	: 冷媒循環量	[kg/h]
D	: 連絡配管外径	[m]
Din	: 連絡配管内径	[m]
Dins	: 断熱材厚さ	[m]
P	: 圧力	[MPa]
H	: エンタルピー	[kJ/kg]
η	: 冷媒粘性係数	[Pa · s]
Cp	: 冷媒比熱	[kJ/kg · K]
λ	: 熱伝導率	[W/m · K]
ρ	: 冷媒密度	[kg/m ³]
α	: 表面熱伝達率	[W/m ² · K]

(添え字)

- ave : 平均値
- in : 流入冷媒
- out : 流出冷媒
- ins : 断熱材

$$\text{Flux} = \frac{Gr}{0.25 \times \pi \times D_{in}^2 \times 3600} \quad \text{式 99}$$

$$P_{ave} = \frac{P_{in} + P_{out}}{2} \quad \text{式 100}$$

$$H_{ave} = \frac{H_{in} + H_{out}}{2} \quad \text{式 101}$$

$$\text{Re} = \frac{\text{Flux} + D_{in}}{\eta_{ave}} \quad \text{式 102}$$

$$\text{Pr} = \frac{\eta_{ave} \times Cp_{ave}}{\lambda_{ave}} \quad \text{式 103}$$

$$\text{Nu} = 0.023 \times \text{Re}^{0.8} \times \text{Pr}^{0.3} \quad (\text{冷媒温度} > \text{周围温度}) \quad \text{式 104}$$

$$\text{Nu} = 0.023 \times \text{Re}^{0.8} \times \text{Pr}^{0.4} \quad (\text{冷媒温度} < \text{周围温度}) \quad \text{式 105}$$

$$\Delta h = \frac{\pi \times L \times \Delta T}{\left(\frac{1}{\text{Nu} \times \lambda} + \frac{D_{ins}}{\lambda_{ins}(D + D_{ins})} + \frac{1}{(D + 2 \times D_{ins}) \times \alpha} \right)} \times \frac{3600}{Gr} \quad \text{式 106}$$

$$\Delta P = \left(\frac{0.3164}{\text{Re}^{0.25}} \times \frac{L}{D_{in}} \times \frac{\text{Flux}^2}{2 \times \rho} \times 2.11676 \pm \text{高低差}(m) \times \rho \times 9.80665 \right) \times \frac{1}{10^6} \quad \text{式 107}$$

$$P_{out} = P_{in} - \Delta P \quad \text{式 108}$$

$$h_{out} = h_{in} + \Delta h \quad \text{式 109}$$

3.3.1.4 室外機の計算

室外機は室外熱交換器や圧縮機といった複数のコンポーネントから成る。現行のビルマルチ空調機の動作を模擬するための計算フローを図 148～図 149 に示す。

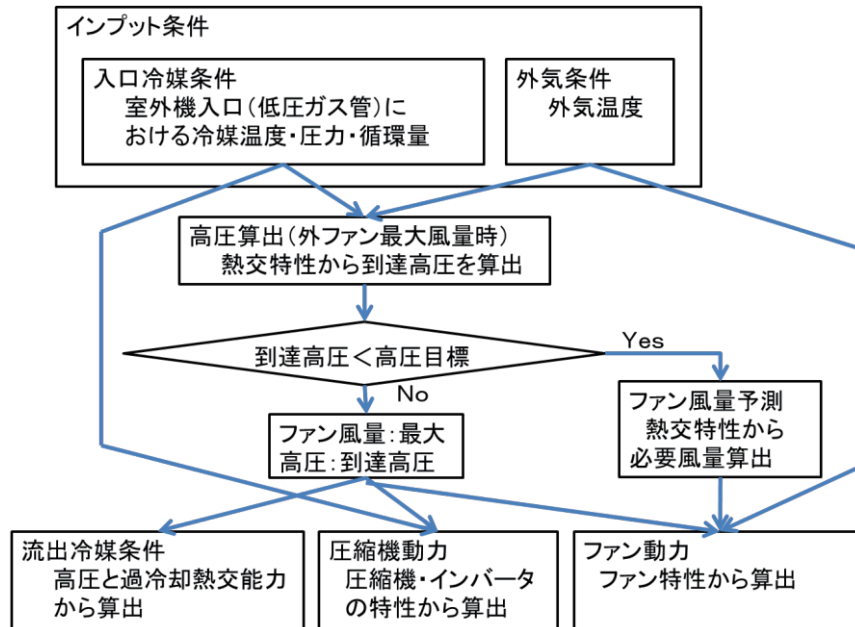


図 148 室外機計算フロー（冷房時）

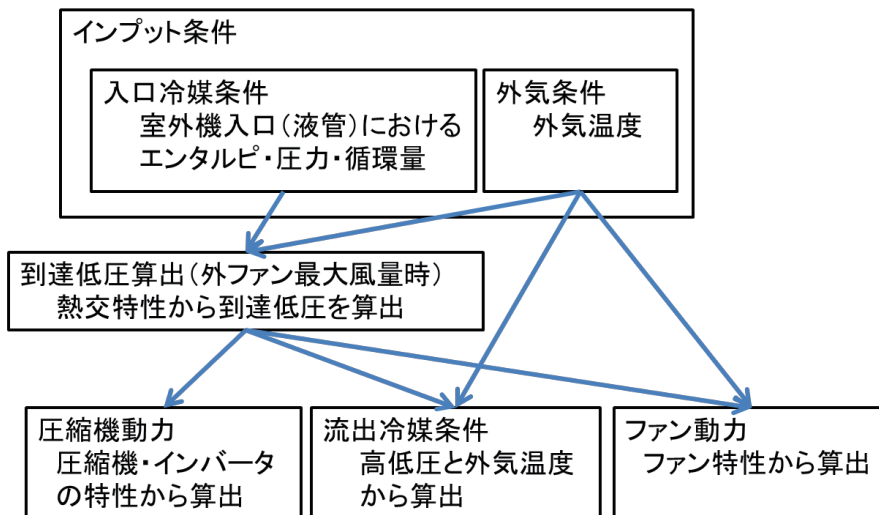


図 149 室外機計算フロー（暖房運転）

(1) 室外熱交換器の計算モデル

室外熱交換器は室内機と同等の計算モデルによる予測が可能である。バイパスファクターモデルによる予測と試験室実験における実測結果との比較を図 150、図 151 に示す。

表 66 室外熱交換機パラメータ例 (10HP 機)

	コイル表面 相対湿度上限	バイパス ファクタ	コイル表面温度－冷媒温度 $\Delta T = a \times SCH^2 + b \times SCH + c$		
			a	b	c
冷房	—	0.00163	- 0.0862	0.8870	0.2270
暖房	98%	0.3444	0	0.1548	0

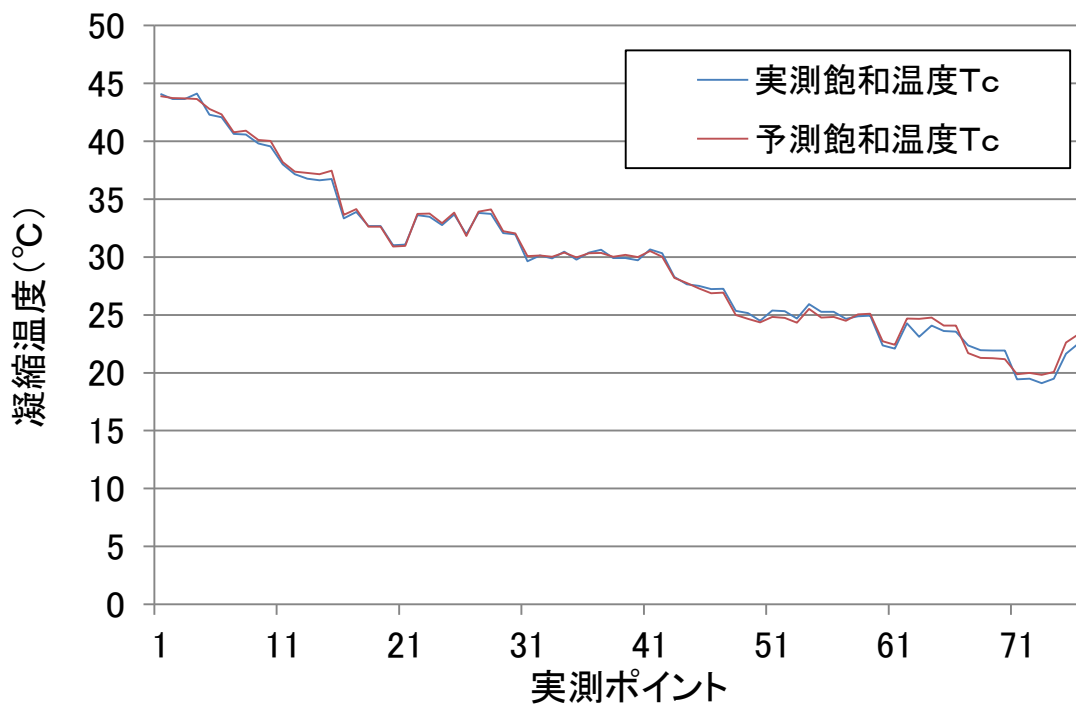


図 150 冷媒凝縮温度予測値と実測値の比較

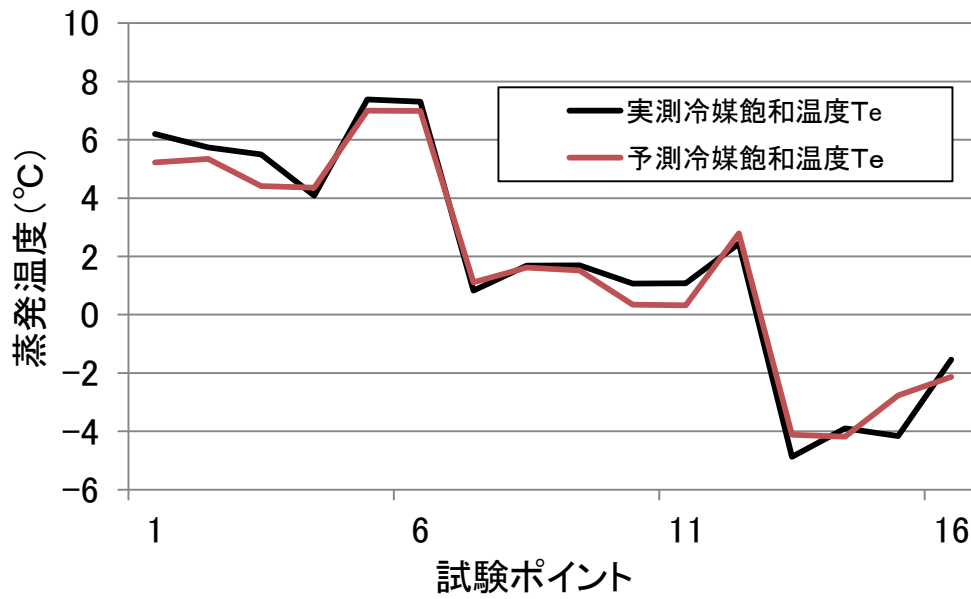


図 151 冷媒蒸発温度予測値と実測値の比較

よってバイパスファクターモデルを用いて室外熱交換器の挙動を予測し、冷媒飽和圧・ファン風量の計算が可能となる。

(2) 圧縮機動力の算出

圧縮機消費電力を、試験結果から凝縮圧力、蒸発圧力、回転数から算出する特性式を作成した。

- Wfre : 圧縮機回転数 [rps]
- Gr : 冷媒循環量 [kg/h]
- ρ_{ref} : 吸入冷媒密度 [kg/m³]
- Vcys : シリンダ容積 [cm³]
- Effvol : 容積効率
- Wcomp : 圧縮機消費電力 [kW]
- Pc : 凝縮圧力 [MPa]
- Pe : 蒸発圧力 [MPa]

$$Wfre = \frac{Gr}{3600} \div \left\{ \rho_{ref} \times \frac{Vcys}{10^6} \times Effvol \right\}$$

式 110

$$Wcomp = \{ WcComp1 \times Pc + WcComp2 \times Pe + WcComp3 \} \times Wfre$$

式 111

表 67 10HP 室外機の係数

WcComp1	WcComp2	WcComp3	Vcys	Effvol
0.0416	-0.0380	0.00625	53.1	0.95

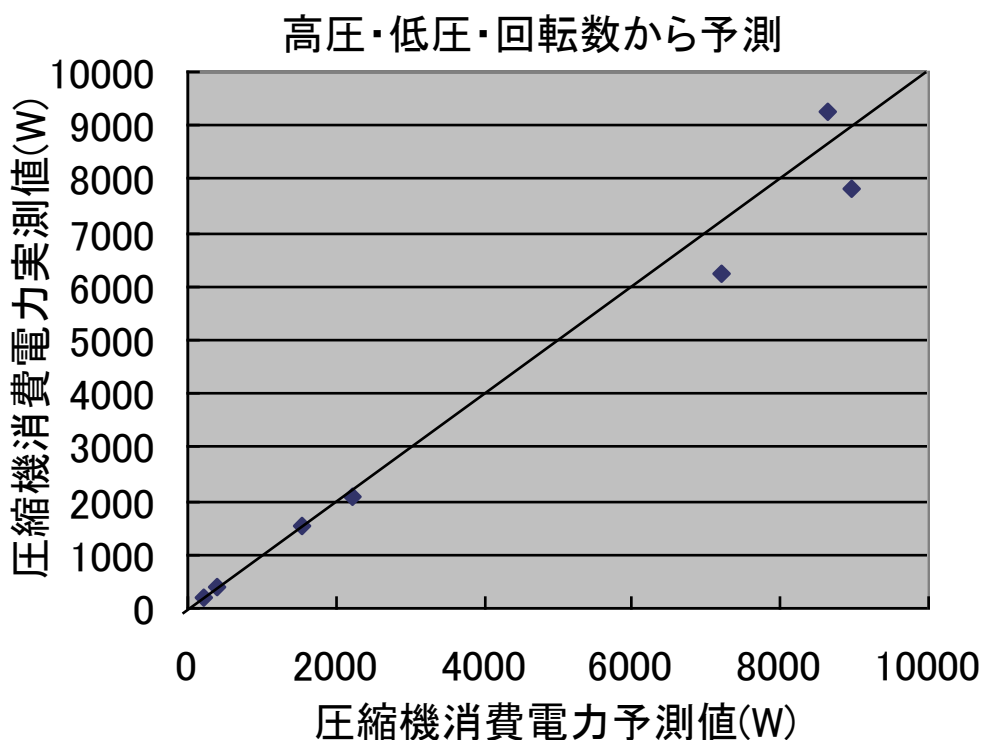


図 152 圧縮機消費電力予測値と実測値比較

ここで、表 67 で示した係数には、圧縮機に付随するインバータ損失による消費電力増大効果分の影響も含まれている。

また、圧縮機動力の計算に関しては、下記のような簡易式でもある程度予測することが可能である。

Wfre	: 圧縮機回転数	[rps]
Gr	: 冷媒循環量	[kg/h]
ρ_{ref}	: 吸入冷媒密度	[kg/m ³]
Vcys	: シリンダ容積	[cm ³]
Effvol	: 容積効率	
Wcomp	: 圧縮機消費電力	[kW]
Pc	: 凝縮圧力	[MPa]
Pe	: 蒸発圧力	[MPa]
H _{ps} (,)	: 比エンタルピー (圧力、比エントロピ) の関数	
H _{tp} (,)	: 比エンタルピー (温度、圧力) の関数	
S _{tp} (,)	: 比エントロピ (温度、圧力) の関数	
η_{inv}	: インバータ損失	
η_{comp}	: 圧縮効率	

$$Gr = \frac{V_{cys}}{10^6} \times W_{fre} \times \rho_{ref} \times Effvol \times 3600$$

式 112

$$W_{comp} = \frac{1}{\eta_{inv}} \times \frac{[H_{ps}\{P_c, S_{tp}(T_s, P_e)\}] - H_{tp}(T_s, P_e)}{\eta_{comp}} \times \frac{Gr}{3600}$$

式 113

尚、圧縮機の挙動については、より詳細な特性データが開示されている圧縮機を用いる場合にはその特性データに沿った計算を行うことが可能である。その場合には、インバータ損失は別途インバータの性能特性を元に想定する必要があることに留意が必要である。

(3) 冷房運転時の冷媒流出温度の算出

多くのメーカーのビルマルチ空調機は、冷房時に室外熱交から流出した冷媒をさらに冷却するための過冷却熱交を備えている。これは連絡配管中に冷媒が圧力損失により液単相流から二相流に遷移するのを防止するためである。

過冷却熱交を持たない室外機の場合、室外機出口冷媒温度は室外熱交出口冷媒温度とほぼ同等の値となるため、高圧冷媒飽和温度と設計・制御上で設定された室外熱交出口過冷却度から容易に求めることが可能であるが、過冷却熱交を持つ場合には過冷却熱交による冷媒温度変化を求める特性式が必要となる。実験室でのビルマルチ挙動試験から得られたデータを元に作成した特性式を以下に示す。

NTLDC	: 液管内フラッシュ防止のために必要な SC	[°C]
NTL	: 液管内フラッシュ防止のために必要な出口冷媒温度	[°C]
NTLtrgt	: 通常運転時の液管出口冷媒温度目標値=25°C	
ToutF	: 通常制御（保護なし）時の到達 SC	[°C]
Tout	: 室外機からの出口冷媒温度	[°C]
Tc	: 凝縮温度	[°C]
Te	: 蒸発温度	[°C]
Ta	: 外気温度	[°C]
Gr	: 冷媒循環量	[kg/h]
Ph	: 高低差対応差圧	[kg/cm ²]
BoutSH	: 過冷却熱交バイパス出口 SH	[°C]

$$NTLSC = NTL_{tc} \times Tc + NTL_{te} \times Te + NTL_{gr} \times Gr + NTL_{ph} \times ph + NTL_c \quad \text{式 114}$$

$$NTL = \min (NTL_{trgt}, Tc - NTLSC) \quad \text{式 115}$$

$$ToutF = Tout_{dt} \times (Tc - Ta) + Tout_{sh} \times BoutSH + Tout_{gr} \times Gr + Tout_c \quad \text{式 116}$$

$$Tout = \max (NTL, Tc - ToutF) \quad \text{式 117}$$

表 68 10HP 室外機の係数

NTL _{tc}	NTL _{te}	NTL _{gr}	NTL _{ph}	NTL _c	Tout _{dt}	Tout _{sh}	Tout _{gr}	Tout _c	BoutSH
0.071	-0.178	0.006	1.514	3.269	0.646	-0.333	0.017	-2.448	5

(4) 暖房運転時の冷媒流出温度の算出

暖房運転時の冷媒出口温度は圧縮機吐出冷媒が室外機から流出するまでにどの程度放熱されるかで決まる。運転状況によって、室外機出口でどの程度放熱されているのか特性を把握した。

ToutSH : 室外機出口における過熱度 [°C]
 Tout : 室内機出口冷媒温度 [°C]
 Te : 蒸発温度 [°C]
 Gr : 冷媒循環量 [kg/h]

$$ToutSH = Tout_{te} \times Te + Tout_{gr} \times Gr + Tout_c$$

式 118

$$Tout = Te + ToutSH$$

式 119

表 69 10HP 室外機の係数

Tout _{te}	Tout _{gr}	Tout _c
-0.453	0.037	-3.872

(5) 室内機の空調ゾーン毎の個別発停による影響予測

室内機がサーモオン、サーモオフすることによって稼働台数が変化することによって運転効率が影響を受ける。そこで、性能予測計算にゾーン毎の発停によって圧縮機最小循環量以下になった場合の冷媒圧力変化の影響の検討を行った。冷房運転時の模式図を図 153～図 154 に示す。図に示すようにゾーン内の負荷が小さく最小容量を下回るとゾーン内空調機の発停が発生する。この場合、発停によって冷媒循環量が大きく変動するため運転状況が大きく変動する。

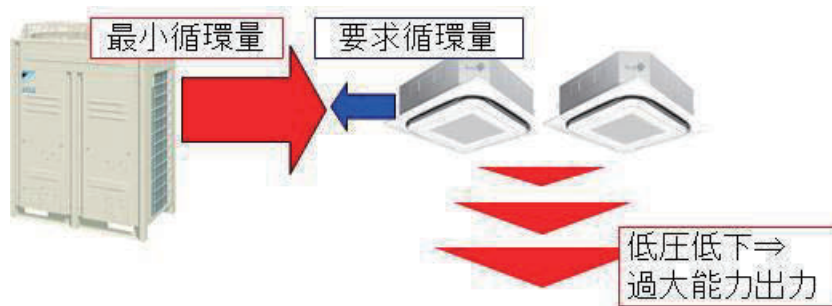


図 153 圧縮機最小循環量制約による低圧低下の模式図

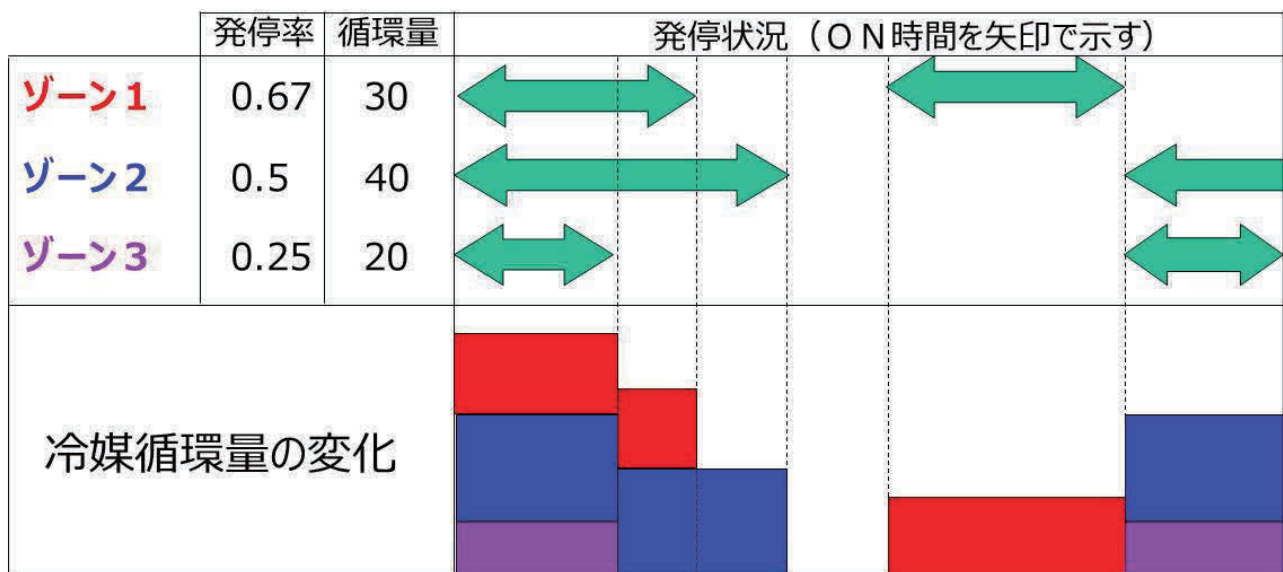


図 154 室内ゾーン個別発停影響予測

室内ゾーン個別発停に併せた冷媒循環量の変動を性能計算に直接組み込むと、計算負荷が何倍にも増加するため、下記のように簡易化して計算モデルの構築を行った。

<計算手順>

1. システム全体の計算は平均冷媒循環量を用いて行う。

(冷房時)

2. その後、発停の状況による低圧の低下幅を算出する。
3. 上記発停による低下も含めた平均低圧を算出する。
4. 低圧のみ3項で計算した値を用いて、システムの消費電力を算出する。

(暖房時)

2. その後、発停の状況による高圧の上昇幅を算出する。
3. 上記発停による上昇も含めた平均高圧を算出する。
4. 高圧のみ3項で計算した値を用いて、システムの消費電力を算出する。

すなわち、発停に伴う循環量変化で上下に変動する分は平均値で計算するが、圧縮機動力下限を下回ることによる圧力変化分は別途計算・補正する。

(I) 発停発生状況（組合せの算出）

各ゾーン計算部から室外機計算部へ、平均冷媒循環量だけでなく、発停率・稼働時冷媒循環量を送る。

- OF_i : ゾーン i の発停率
- GR_i : ゾーン i 運転中の冷媒循環量
- GR_{iave} : ゾーン i の平均冷媒循環量
- Li : ゾーン i の要求負荷
- Q_{i min} : ゾーン i の風量最小時に出力される能力

$$GR_{iave} = OF_i \times GR_i \quad \text{式 120}$$

$$OF_i = \frac{Li}{Q_{i min}} \quad \text{式 121}$$

各ゾーンの発停による発停率（発停の組合せ）の出現率は、

OF_j : ある発停状況時の発生率（j はある発停状況を示す添え字とする）

$$OF_j = \prod_{i \text{ (運転中のゾーン)}} OF_i \times \prod_{i \text{ (停止中のゾーン)}} (1 - OF_i) \quad \text{式 122}$$

(II) 発停発生状況による冷媒温度（蒸発温度、凝縮温度）の計算

最低循環量による限定により、冷媒温度が変化する量（冷房時の蒸発温度低下、暖房時の凝縮温度上昇）を計算する。

- GRR : 循環量率（最小循環量を割り込む比率）
- Gr_{min} : 圧縮機最小冷媒循環量

$$GR_j = \sum_{i \text{ (稼働中のゾーン)}} GR_i \quad \text{式 123}$$

$$GRR_j = \frac{GR_j}{Gr_{min}} \quad (\text{但し } 0 \leq GRR_j \leq 1) \quad \text{式 124}$$

(冷房運転時)

Tr	: 室内乾球温度	[°C]
Tp	: 室内露点温度	[°C]
ΔH1	: 室内露点温度における飽和空気と冷媒蒸発温度における飽和空気のエンタルピー差	[kJ/kg]
ΔHS1	: 室内露点温度における飽和空気と冷媒蒸発温度における飽和空気の顕熱エンタルピー差	[kJ/kg]

$$R = \Delta HS1 / \Delta H1$$

式 125

$$\text{Tr} - \frac{\text{Tr} - \text{Te}}{\text{GRRj}} > \text{Tp} \text{ 時}$$

$$\text{Tej} = \text{Tr} - \frac{\text{Tr} - \text{Te}}{\text{GRRj}}$$

式 126

それ以外

$$\text{Tej} = \text{Tp} - \left\{ \frac{\text{Tr} - \text{Te}}{\text{GRRj}} - (\text{Tr} - \text{Tp}) \right\} \div R$$

式 127

(暖房運転時)

$$\text{Tcj} = \text{Ta} + \frac{\text{Tc} - \text{Ta}}{\text{GRRj}}$$

式 128

(III) 冷媒温度（蒸発温度、凝縮温度）平均値の計算

冷媒温度（蒸発温度、凝縮温度）の平均値を算出し、圧縮機動力の計算に用いる。

(冷房時)

$$\text{Te}_{ave} = \sum_1^j (\text{OFj} \times \text{Tej})$$

式 129

(暖房時)

$$\text{Tc}_{ave} = \sum_1^j (\text{OFj} \times \text{Tcj})$$

式 130

3.3.2 ビルマルチエアコンの特性式妥当性検証結果

3.3.2.1 室内機特性式の評価（冷房：時間平均データを使用した場合）

前述したように、ビルマルチエアコンの潜熱処理量を評価するためには、室内機の運転予測が重要である。そこでまず、室内機特性式の精度評価を行う。

Aビルで室内機の吸込み・吹出し温湿度実測結果から算出した実測能力と、運転中の冷媒状態（蒸発飽和温度、SH）をもとに特性式を用いて算出した室内機能力の比較により、特性式の精度評価を行った。

精度評価の対象としては、全熱能力と顕熱能力比（SHF：Sensible Heat Factor）を用いる。ここでSHFとは以下の式で表される値である。

$$SHF = \frac{Q_{sen}}{Q_{total}}$$

式 131

(1) FXYMP71AA (天井埋め込みダクト型、7.1kW 機)

当該室内機については、以前試験室にて行った実測結果から作成した特性式パラメータを用いて精度評価を実施した。

表 70 FXYMP71AA の特性パラメータ(冷房)

BF	a	b	c
0.1608	0	0.5973	0

従来のバイパスファクターモデルで予測した結果を以下に示す。

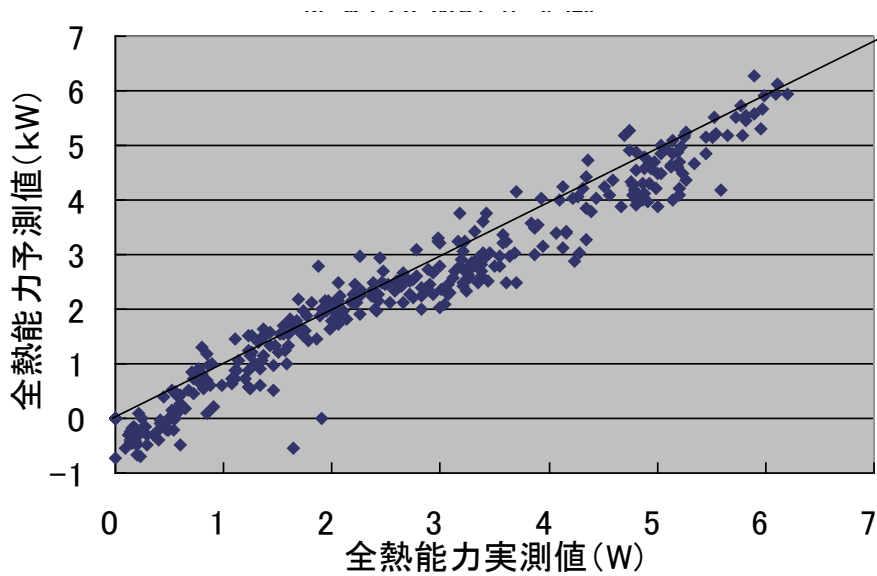


図 155 従来のバイパスファクターモデルによる全熱能力予測精度

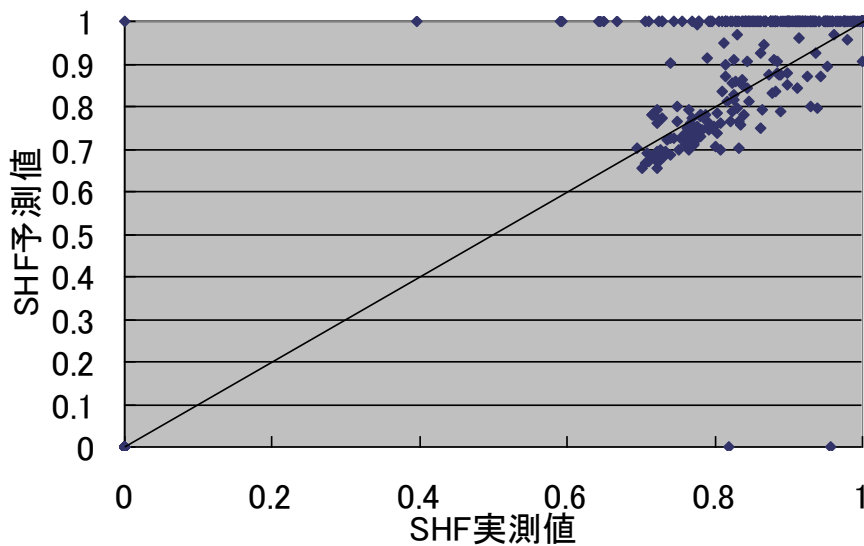


図 156 従来のバイパスファクターモデルによる SHF 予測精度

次に、修正バイパスファクターモデルを用いた場合の予測結果を以下に示す。

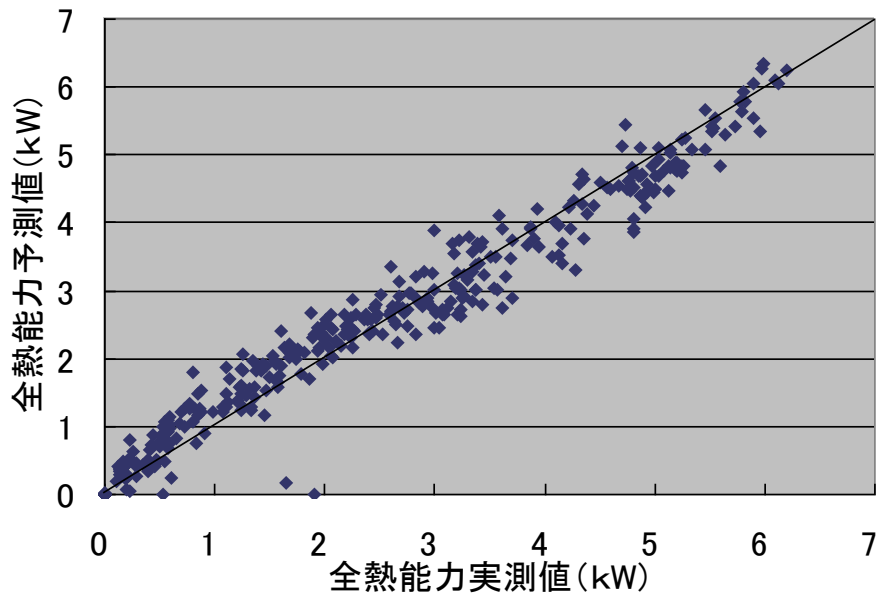


図 157 修正バイパスファクターモデルによる全熱能力予測精度

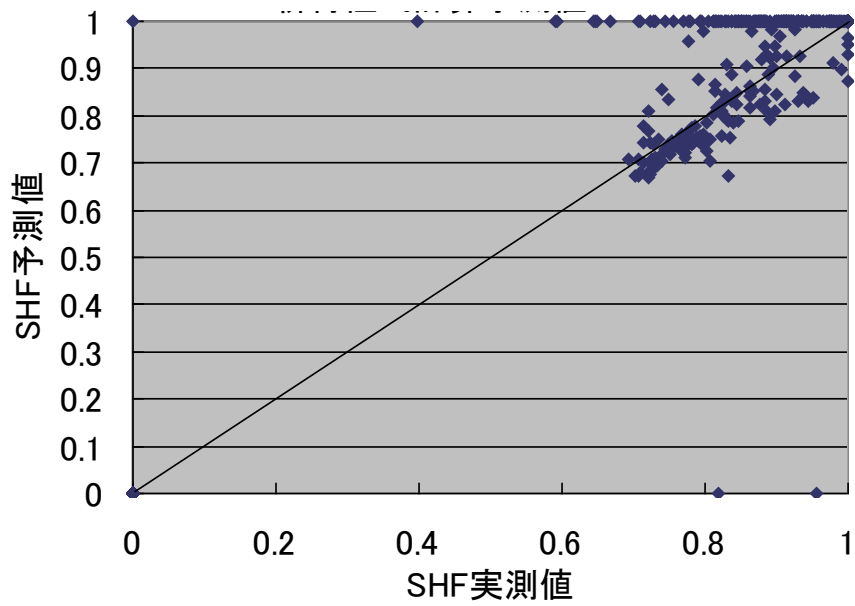


図 158 修正バイパスファクターモデルによる SHF 予測精度

図 155, 図 157 から明らかなように、従来のバイパスファクターモデル、修正バイパスファクターモデルともに、全熱能力を概ね予測出来ているが、両図を比較すると、従来のバイパスファクターモデルの方が予測値とばらつきが多少大きく、一部ではマイナスの能力を予測している。ばらつきの縮小は熱交表面温度を SH の二次式で表したことの効果であり、マイナス値を予測するような全体傾向としての乖離は特性式パラメータが室内機によらず共通であることに起因すると思われる。

今回の試験機では、制御仕様上室内機の SH 目標値を大きく変化させるような制御を導入していない為、従来のバイパスファクターモデルと修正バイパスファクターモデルの差異は小さいが、制御仕様によっては SHF も大きく変化させて制御する場合も想定され、そういった場合には修正バイパスファクターモデルの採用による精度改善は重要なポイントであると考えられる。

次に図 156, 図 158 に示す SHF の予測精度であるが予測モデルによって大きな差は見られなかった。一方で、予測と実測で大きく乖離のあるデータが多数生じている。この乖離の原因に関しては、1 時間分のデータを平均して収集していることが原因であると考えられる。1 時間平均を行うことで、蒸発温度が実運転上よりも高いものとして計算されており、除湿性能が無い（即ち SHF = 1）という予測値を算出しているものと思われる。

そこで、収集データのうち、膨張弁開度データから連続稼働していると思われるポイントのみを抽出し、予測値と実測値を比較したデータを下記に示す。

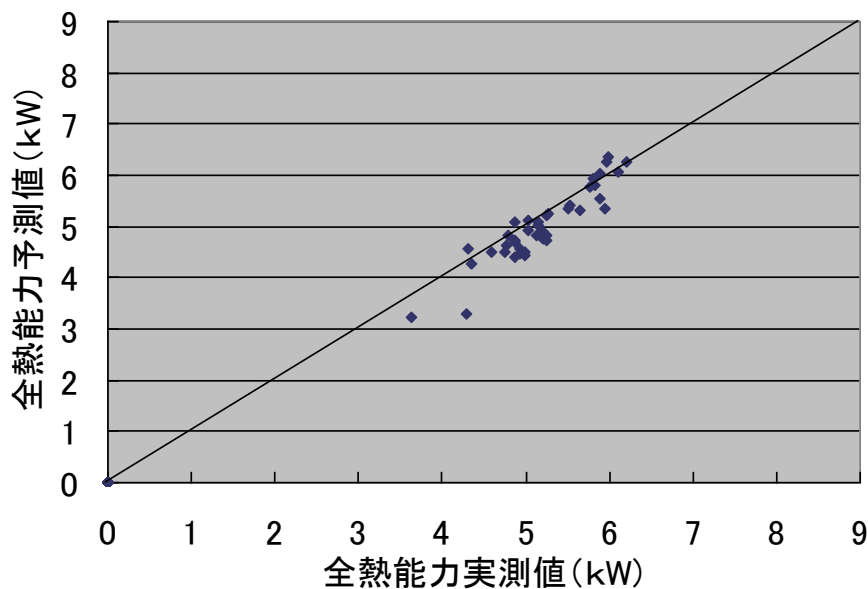


図 159 修正バイパスファクターモデルによる全熱能力予測精度（連続稼働時）

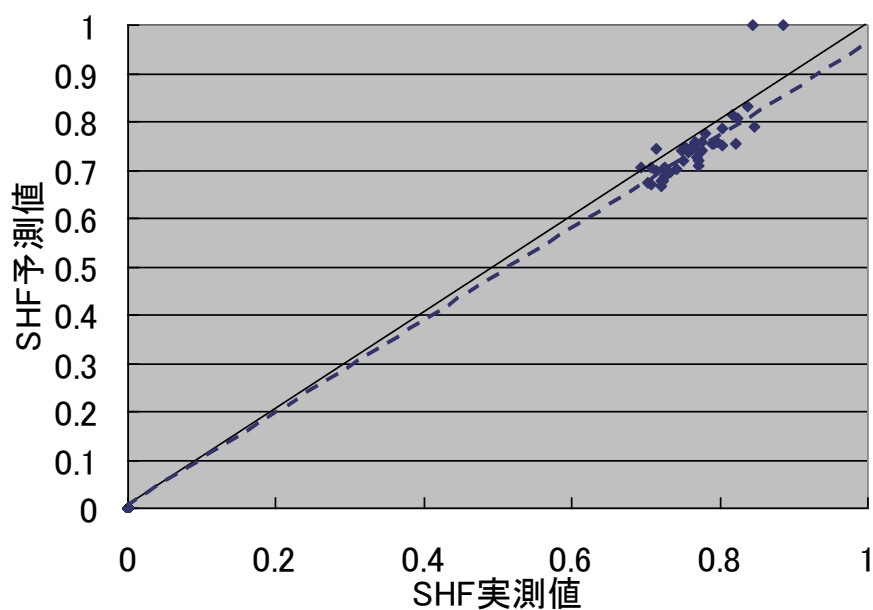


図 160 修正バイパスファクターモデルによる SHF 予測精度 (連続稼働時)

図 160 からわかるように、稼働時のデータでは SHF も高い精度で予測出来ている。

(2) FXYMP90AA (天井埋め込みダクト型、9.0kW 機)

当該室内機に関しては事前に実測・作成した特性式パラメータが存在しないため、本試験のデータから特性式パラメータを作成した。

表 71 FXYMP90AA の特性パラメータ(冷房)

BF	A	b	c
0.055	0	0.71	0.44

従来のバイパスファクターモデルで予測した結果を以下に示す。

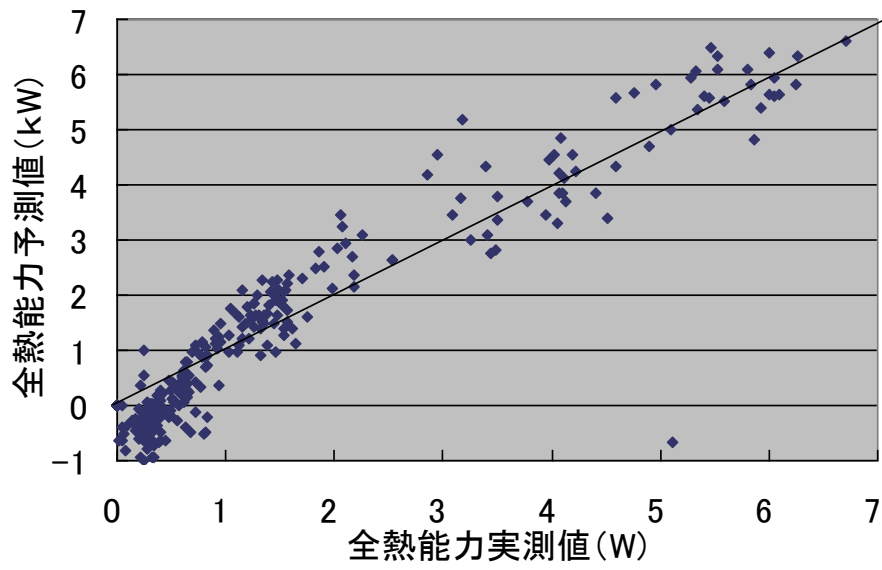


図 161 従来のバイパスファクターモデルによる全熱能力予測精度

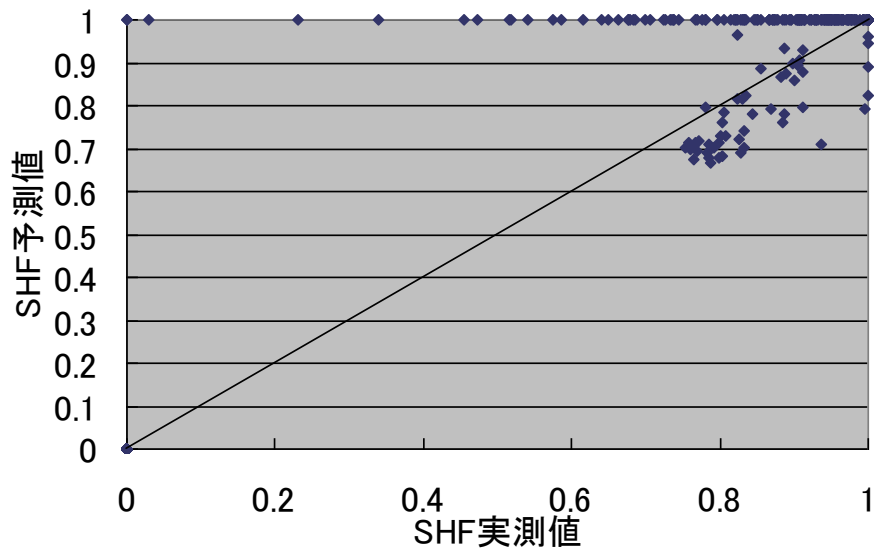


図 162 従来のバイパスファクターモデルによる SHF 予測精度

次に、修正バイパスファクターモデルを用いた場合の予測結果を以下に示す。

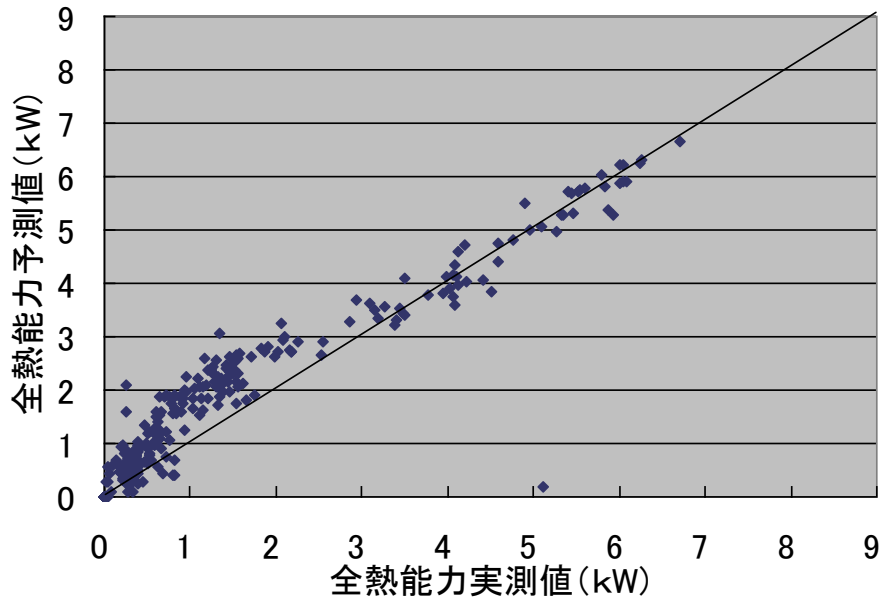


図 163 修正バイパスファクターモデルによる全熱能力予測精度

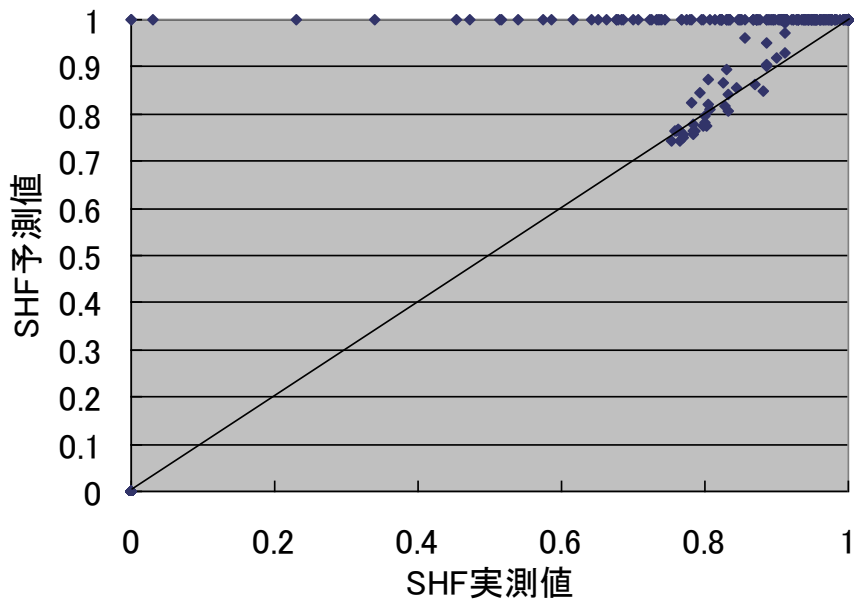


図 164 修正バイパスファクターモデルによる SHF 予測精度

図 161、図 163 の比較から、本 9.0kW 室内機においても、先述の 7.1kW 室内機同様に全熱能力を概ね予測出来ており、従来のバイパスファクターモデルの方が予測値とばらつきが多少大きく、一部ではマイナスの能力を予測している。上述の通り、修正バイパスファクターモデルの特性式パラメータは今回の試験データを元に作成しているが、図 163 の結果から、室内機が変わっても特性式パラメータさえ入れ替えれば、室内機特性を表すことが可能なモデルとなっていることが確認できる。

次に図 162、図 164 に示す SHF の予測精度に関しても、7.1kW 室内機と同様に、予測と実測で大きく乖離のあるデータが多数生じている。ここでも、収集データのうち、膨張弁開度データから連続稼働していると思われるポイントのみを抽出し、予測値と実測値を比較したデータを下記に示す。

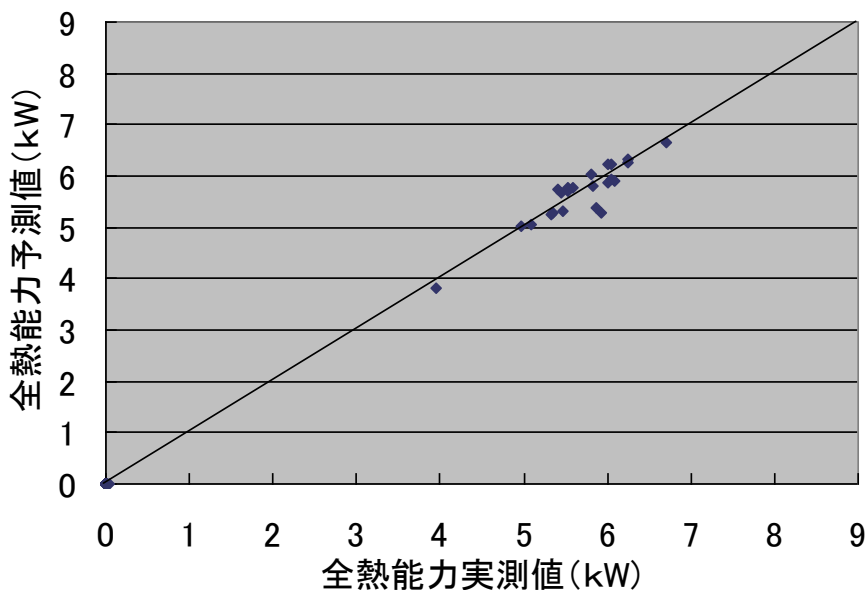


図 165 修正バイパスファクターモデルによる全熱能力予測精度（連続稼働時）

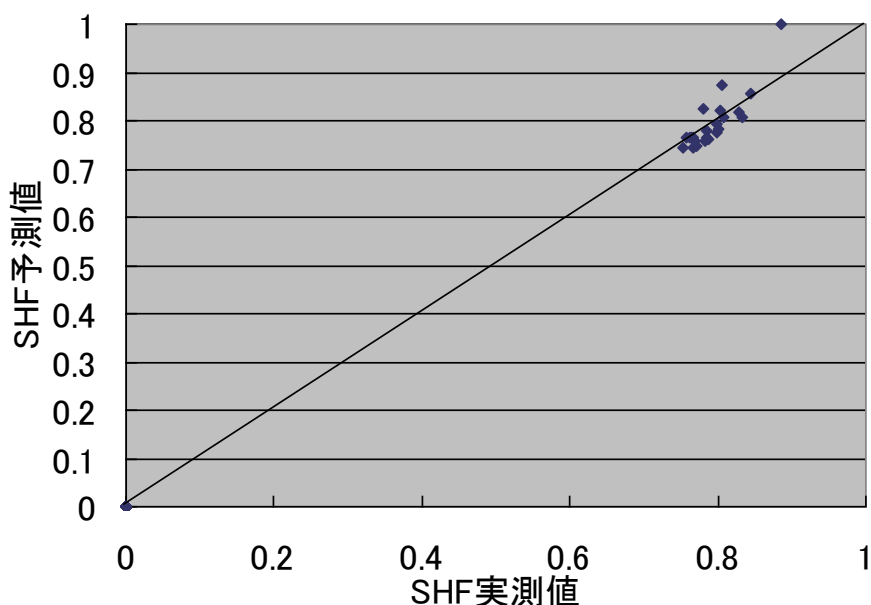


図 166 修正バイパスファクターモデルによる SHF 予測精度（連続稼働時）

図 166 からわかるように、稼働時のデータでは SHF も高い精度で予測出来ている。

(3) FX YMP140AA (天井埋め込みダクト型、14.0kW 機)

当該室内機に関しても事前に実測・作成した特性式パラメータが存在しないため、本試験のデータから特性式パラメータを作成した。

表 72 FX YMP140AA の特性パラメータ(冷房)

BF	A	b	C
0.22	-0.048	0.67	1.18

従来のバイパスファクターモデルで予測した結果を以下に示す。

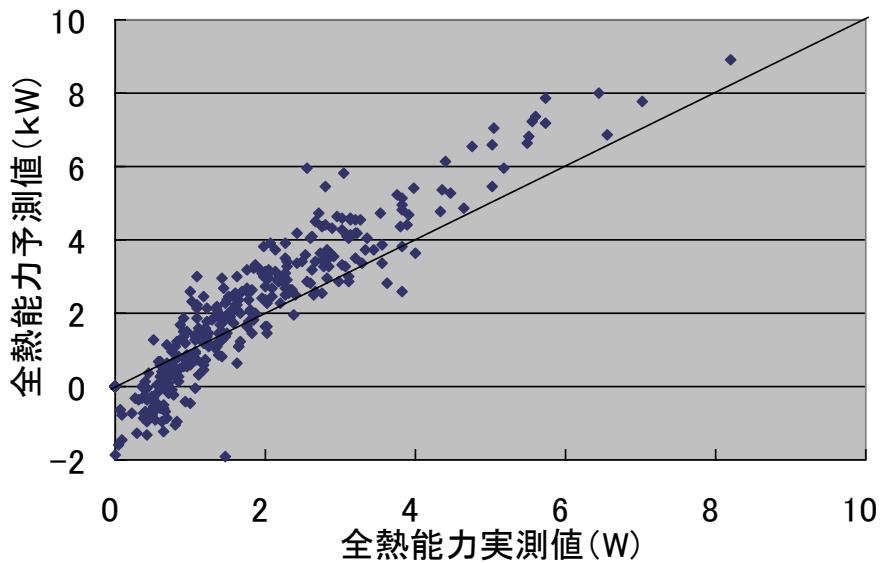


図 167 従来のバイパスファクターモデルによる全熱能力予測精度

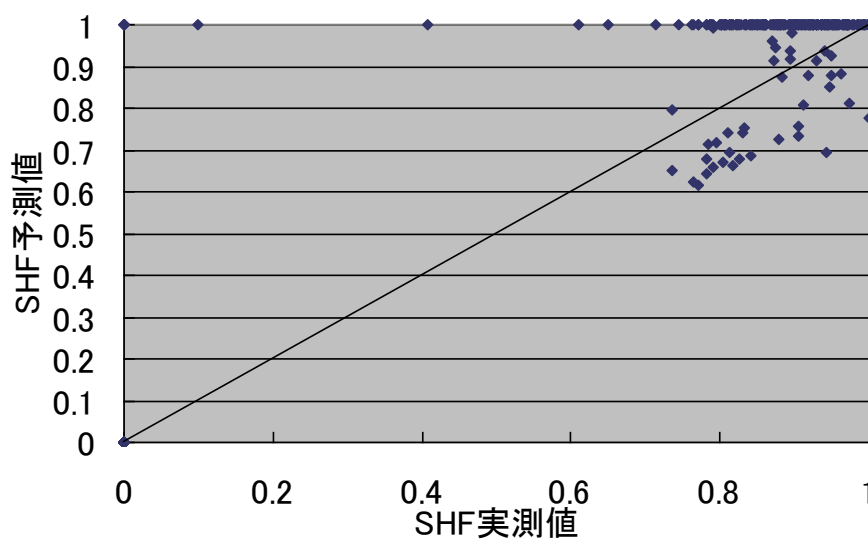


図 168 従来のバイパスファクターモデルによる SHF 予測精度

次に、修正バイパスファクターモデルを用いた場合の予測結果を以下に示す。

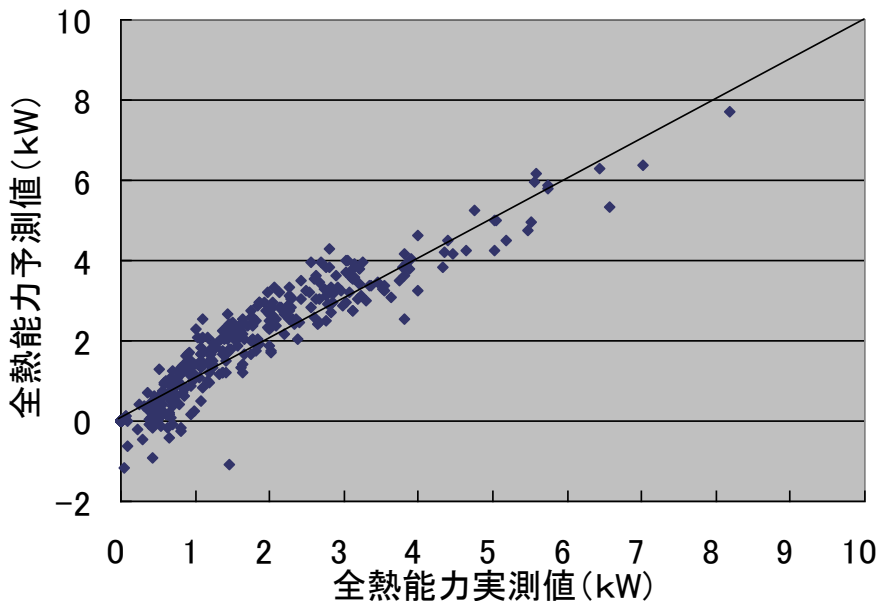


図 169 修正バイパスファクターモデルによる全熱能力予測精度

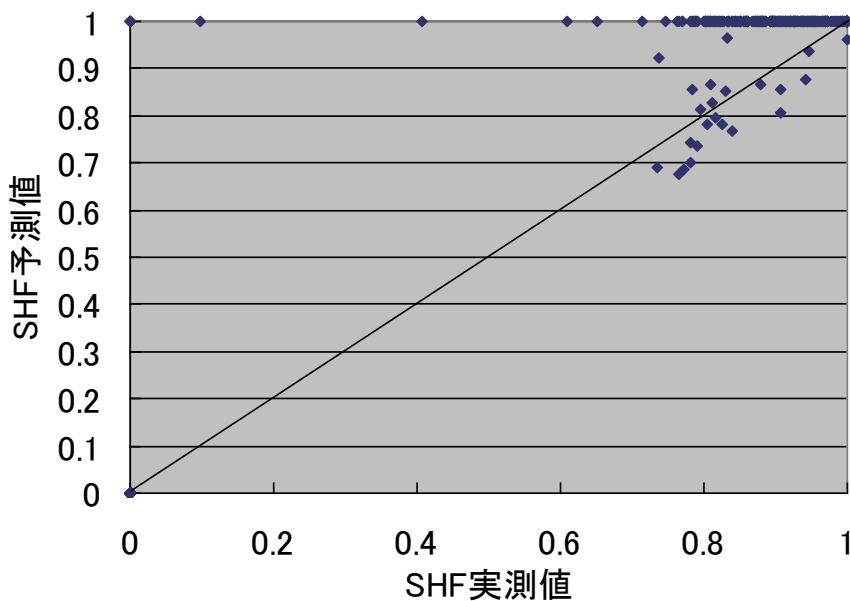


図 170 修正バイパスファクターモデルによる SHF 予測精度

図 167、図 169 の比較から、本 14.0kW 室内機においても、先述の 7.1kW 室内機同様に全熱能力を概ね予測出来ており、従来のバイパスファクターモデルの方が予測値とばらつきが多少大きく、特に低負荷領域でマイナスの能力を予測している。上述の通り、修正バイパスファクターモデルの特性式パラメータは今回の試験データを元で作成しているが、図 169 から、室内機が変わっても特性式パラメータさえ入れ替えれば、室内機特性を表すことが可能なモデルとなっていることが確認できる。

次に図 168、図 170 に示す SHF の予測精度に関しても、7.1kW 室内機と同様に、予測と実測で大きく乖離のあるデータが多数生じている。ここでも、収集データのうち、膨張弁開度データから連続稼働していると思われるポイントのみを抽出し、予測値と実測値を比較したデータを下記に示す。

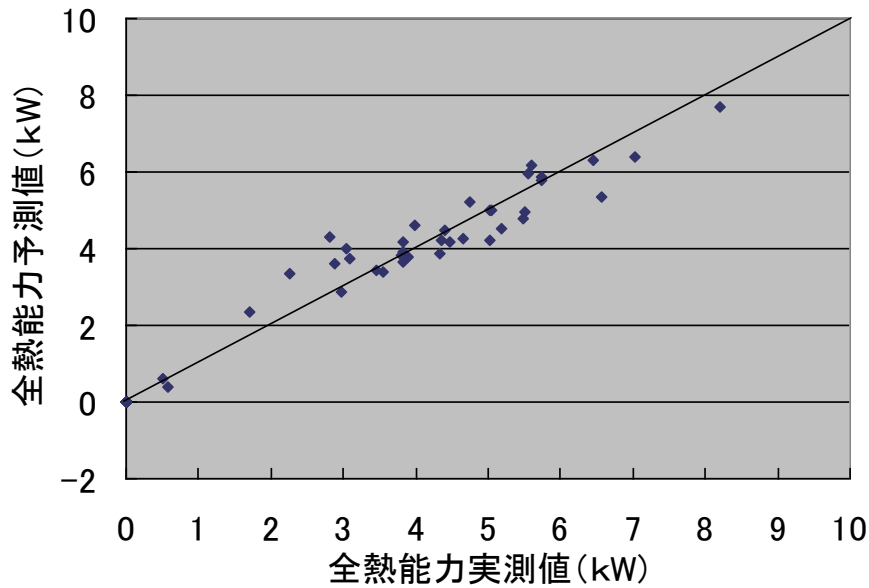


図 171 修正バイパスファクターモデルによる全熱能力予測精度（連続稼働時）

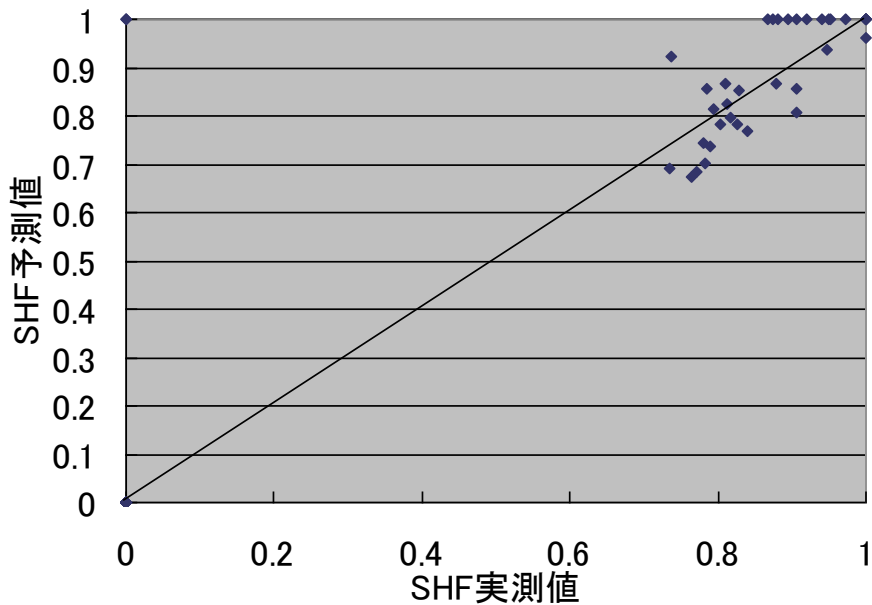


図 172 修正バイパスファクターモデルによる SHF 予測精度（連続稼働時）

図 172 からわかるように、予測値に多少のばらつきは見られるものの、稼働時のデータでは SHF も概ね予測出来ている。

上記のように、修正バイパスファクターモデルを用いた計算により室内機の全熱能力特性・SHF 特性を比較的高精度に予測できることが分かった。

また本評価による重要な知見として、全熱能力は 1 時間平均データを用いて計算してもほぼ正確に予測可能であるのに対し、SHF は発停している時間帯のデータを平均して計算すると実測値と予測値の間に大きな乖離が生じる傾向が見られた。除湿量は蒸発冷媒飽和温度に大きく影響されるので当然ではあるが、計測されたデータを用いてシミュレーションを行う場合、そのデータが瞬時値か、ある時間範囲の平均値かに留意して計算することが必要であるものとする。

3.3.2.2 室内機特性式の評価（暖房：時間平均データを使用した場合）

ここでは、Bビルに導入した室内機：FXYMP45AA（天井埋め込みダクト型、4.5kW 機）、FXYMP112AA（天井埋め込みダクト型、11.2kW 機）についてそれぞれ精度評価を実施する。

Bビルで室内機の吸込み・吹出し温湿度実測結果から算出した実測能力と、運転中の冷媒状態（蒸発飽和温度、SC）をもとに特性式を用いて算出した室内機能力の比較により、特性式の精度評価を行った。

精度評価の対象としては、全熱能力を用いる。

(1) FXYMP45AA（天井埋め込みダクト型、4.5kW 機）

当該室内機に関しては事前に実測・作成した特性式パラメータが存在しないため、本試験のデータから特性式パラメータを作成した。

表 73 FXYMP45AA の特性パラメータ(冷房)

BF	A	b	c
0.136	0	0.4844	0

従来のバイパスファクターモデルで予測した結果を以下に示す。

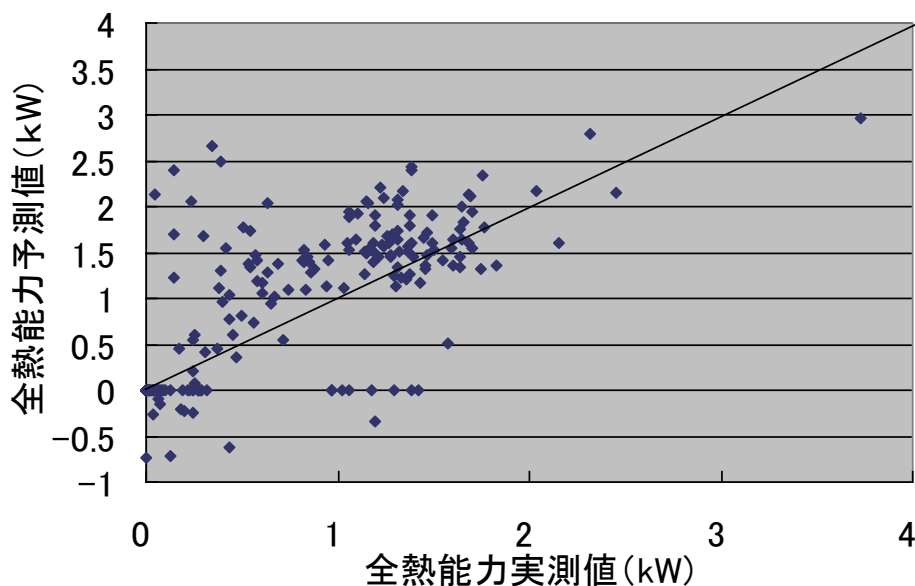


図 173 従来のバイパスファクターモデルによる全熱能力予測精度

次に、修正バイパスファクターモデルを用いた場合の予測結果を以下に示す。

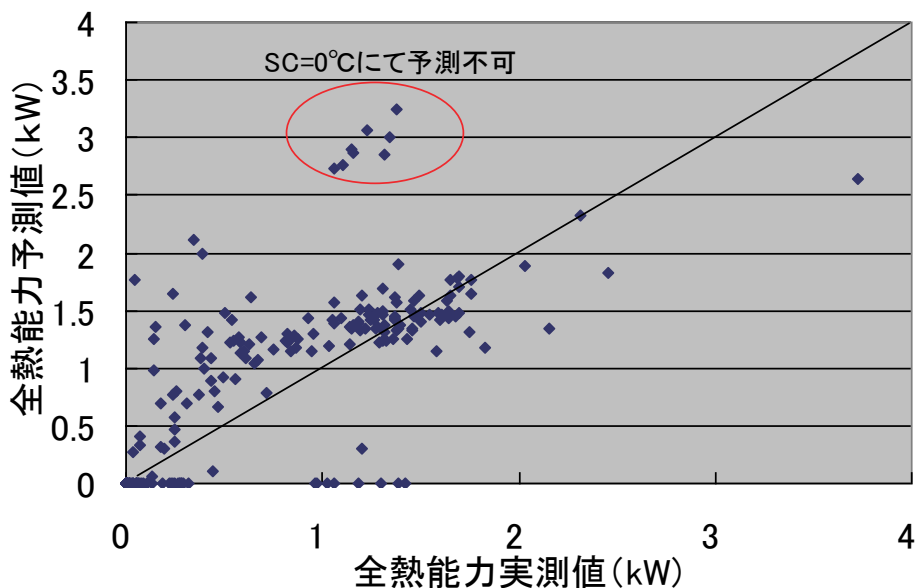


図 174 修正バイパスファクターモデルによる全熱能力予測精度

図 173、図 174 から明らかなように、従来のバイパスファクターモデル、修正バイパスファクターモデルともに、全熱能力を概ね予測出来ているが、両図を比較すると、従来のバイパスファクターモデルの方が予測値のばらつきが多少大きく、修正バイパスファクターモデルの導入による効果が確認された。

尚、修正バイパスファクターモデルによる冷房と暖房の能力予測精度確認結果である図 169 と図 174 を比較すると冷房時の方が、収束性が良い。この傾向は試験室にて室内機を運転した場合の特性式精度評価結果と逆になっている（試験室データで特性式精度評価を行うと、暖房時の方が、収束性が良い）。試験室による評価では、冷房は潜熱能力と顕熱能力の両方が存在し、特に潜熱能力が運転条件に影響されやすいため、予測が難しいものと考えられる。一方で今回のような実測評価においては、暖房時の室内機がサーモオフ時でも冷媒が少量は流れており、若干ながら暖房能力も出力していることや、運転中の除霜運転や連絡配管中にたまった冷媒の回収運転などが冷房時と異なり一旦逆サイクル（冷房サイクル）で運転を行っていることにより、今回のようにある程度の時間平均データを用いた評価ではどうしてもデータそのもののバラつきが大きくなってしまふものと考えられる。

また、図 174 に示したように、今回の特性式精度評価においては一部 SC がついていない状態での運転データが混入しており、適正な能力予測が出来ていない場合があった。このような状態は冷媒充填量が不足していると発生しやすいため、適正な評価のためには適正な施工も重要であることが分かる。収集データのうち、膨張弁開度データから稼働率が高いと思われるポイントのみを抽出し予測値と実測値を比較したデータを下記に示す。

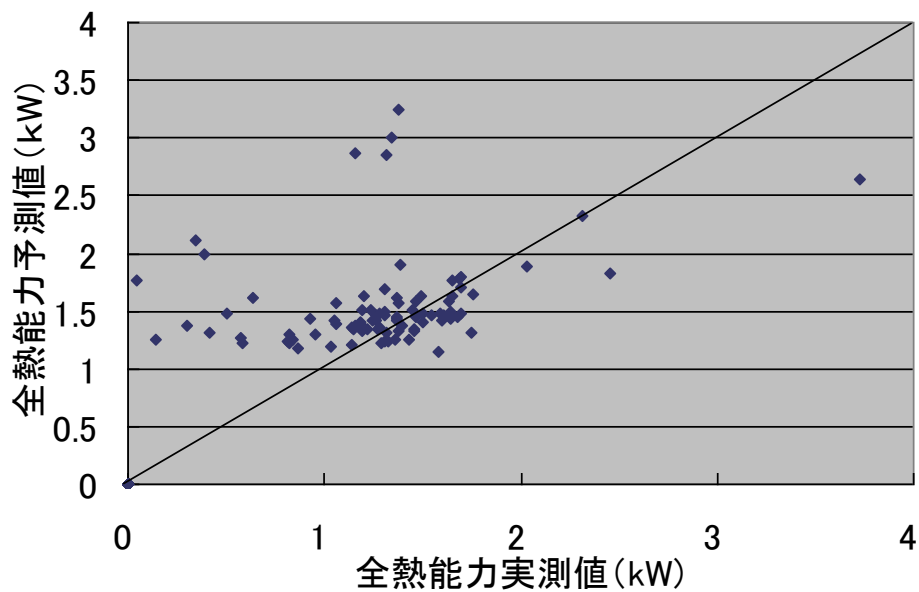


図 175 修正バイパスファクターモデルによる全熱能力予測精度（連続稼働時）

稼働率が高い状態での評価では、収束性に大幅な改善が見られ、発停などによる影響が予測精度に大きな影響を与えることがわかる。

(2) FX YMP112AA（天井埋め込みダクト型、11.2kW 機）

当該室内機に関しては事前に実測・作成した特性式パラメータが存在しないため、本試験のデータから特性式パラメータを作成した。

表 74 FX YMP45AA の特性パラメータ(冷房)

BF	A	b	c
0.08	0	0.30	0.04

従来のバイパスファクターモデルで予測した結果を以下に示す。

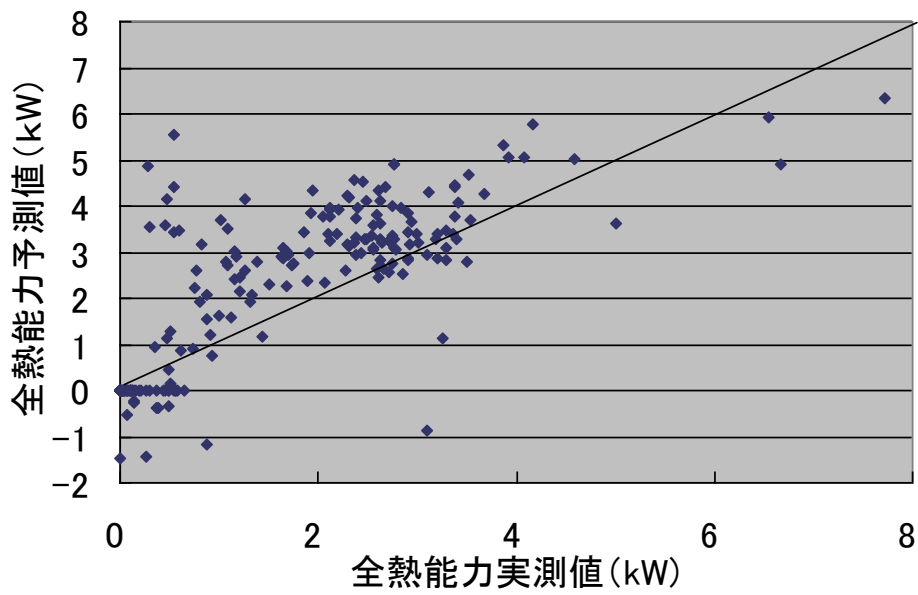


図 176 従来のバイパスファクターモデルによる全熱能力予測精度

次に、修正バイパスファクターモデルを用いた場合の予測結果を以下に示す。

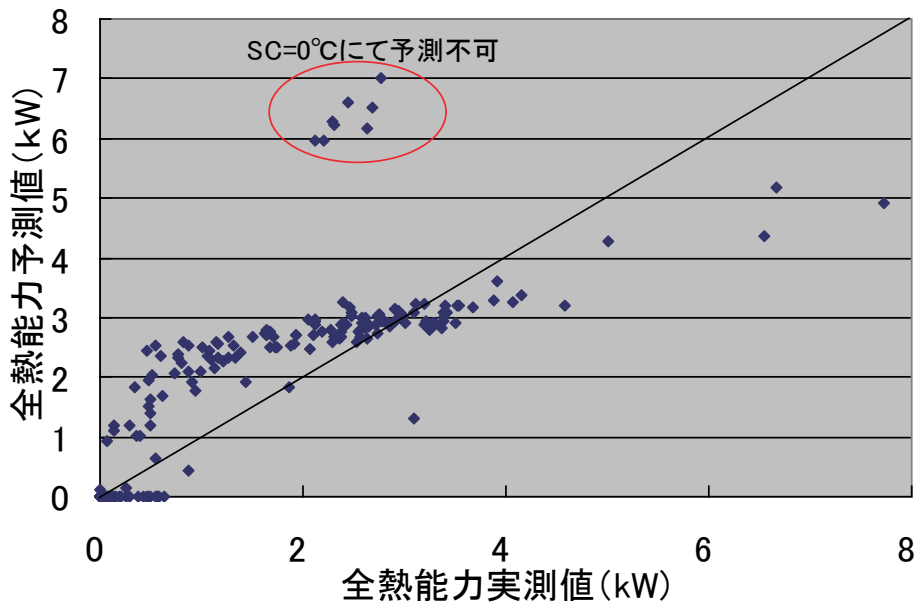


図 177 修正バイパスファクターモデルによる全熱能力予測精度

図 176、図 177 から明らかなように、従来のバイパスファクターモデル、修正バイパスファクターモデルともに、全熱能力を概ね予測出来ているが、両図を比較すると、従来のバイパスファクターモデルの方が予測値のばらつきが大きく、修正バイパスファクターモデルの導入による効果が確認された。

尚、ここでも図 177 からわかるように、暖房時の特性式精度は冷房時よりも低い。

また、図 177 においても、今回の特性式精度評価においては一部 SC がついていない状態での運転データが混入しており、適正な能力予測が出来ていない場合があった。

収集データのうち、膨張弁開度データから稼働率が高いと思われるポイントのみを抽出し、予測値と実測値を比較したデータを下記に示す。

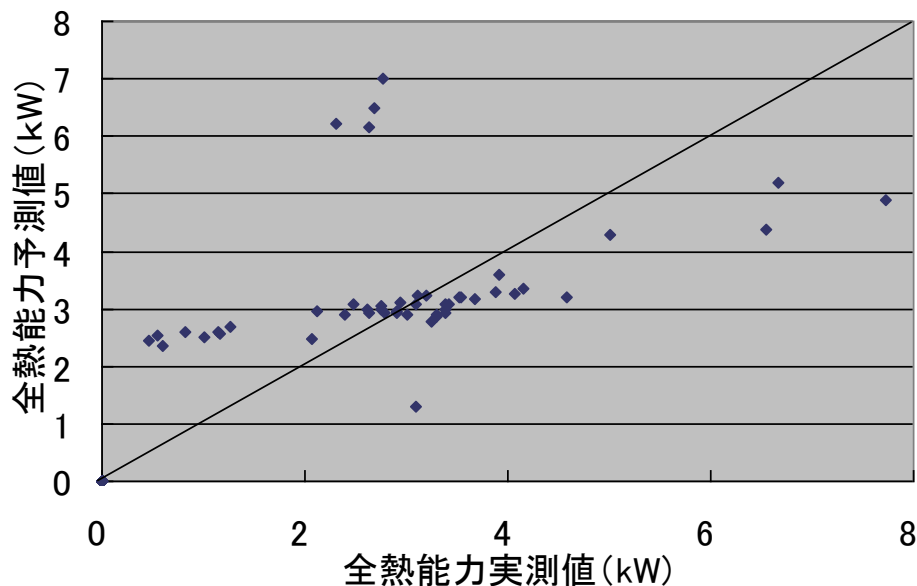


図 178 修正バイパスファクターモデルによる全熱能力予測精度（連続稼働時）

稼働率が高い状態での評価では、収束性に大幅な改善が見られ、発停などによる影響が予測精度に大きな影響を与えることがわかる。

3.3.2.3 室内機特性式の評価（1分毎の瞬時データを使用した場合）

3.3.2.1 では室内機特性式の妥当性を時間平均データを用いて評価した。その結果、平均した時間（1時間）の間運転が継続している状態ではほぼ正確に予測出来ているが、発停などで停止時間が含まれた場合には、全熱能力は予測できるものの SHF は適切に予測できなかった。

そこで、Aビルのデータ収集システムにより短時間間隔でのデータ収集装置を増設し、瞬時データを用いた評価における室内機特性式の有効性を確認した結果を以下に示す。

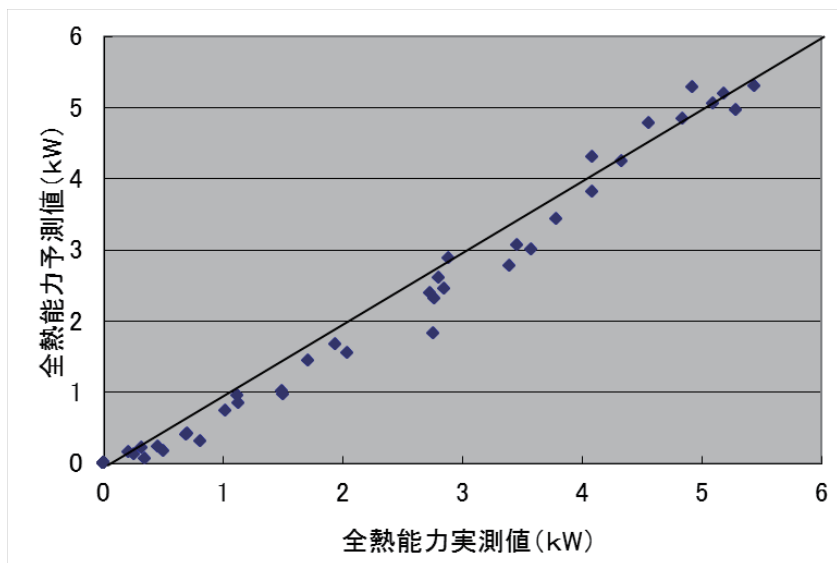


図 179 室内機全熱能力予測精度（冷房、瞬時データ使用、Aビル7.1kW機）

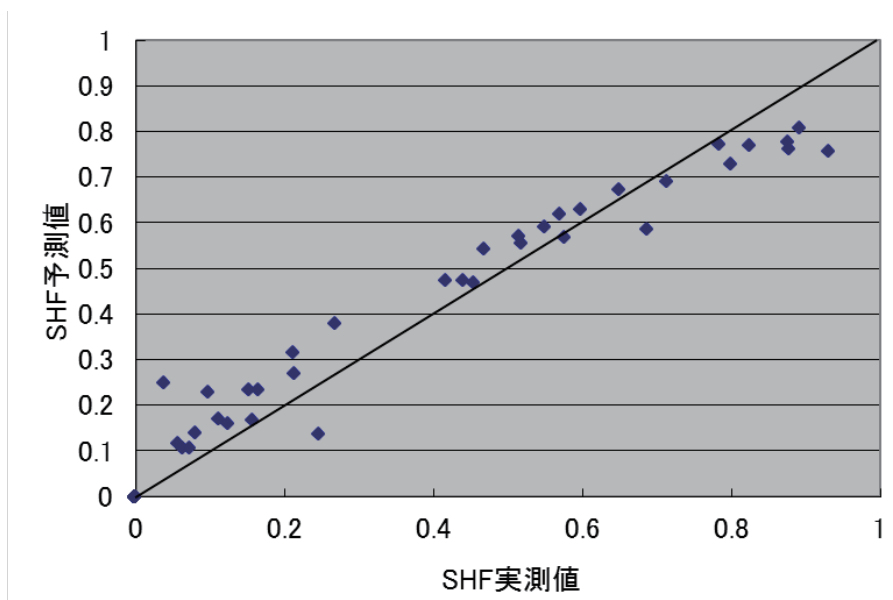


図 180 室内機SHF予測精度（冷房、瞬時データ使用、Aビル7.1kW機）

図 158 で示したように、時間平均データでは、発停時に蒸発温度が鈍るため、SHF を正確に予測出来ない場合があるが、図 179、図 180 からわかるように、冷房時に瞬時データを元にした予測では機器能力、SHF（顕熱比）ともに正確に予測出来ているため、特性式としてはこれで完成とする。

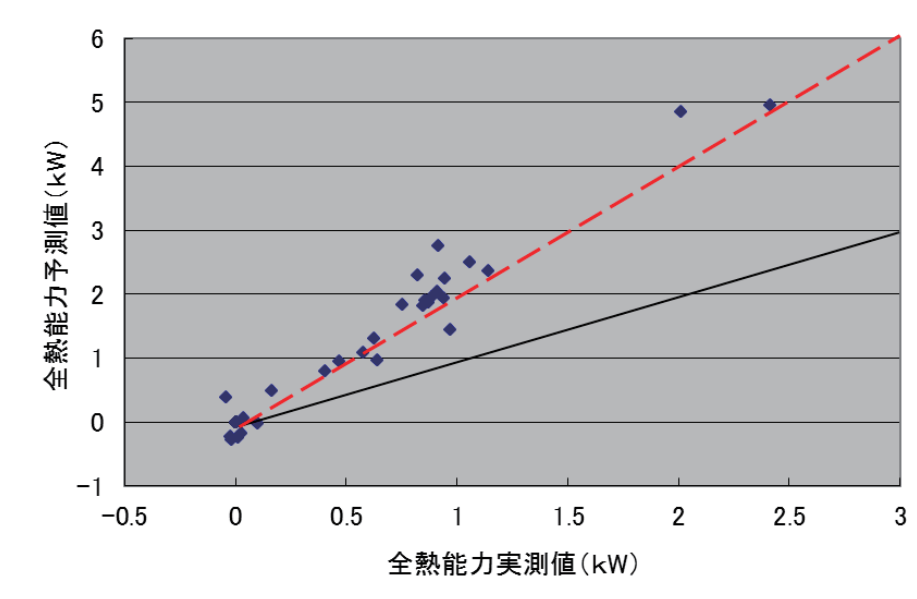


図 181 室内機全熱能力予測精度（暖房、瞬時データ使用、Aビル 14.0kW機）

次に暖房運転の瞬時データによる評価であるが、室内機サーモオン中の瞬時データで比較すると図 181 からわかるようにならかなり予測値の方が大きな値を算出している。この時の室外機の運転状態を詳細に確認すると、暖房継続時間 5～6 分の発停運転を繰り返していることがわかった。

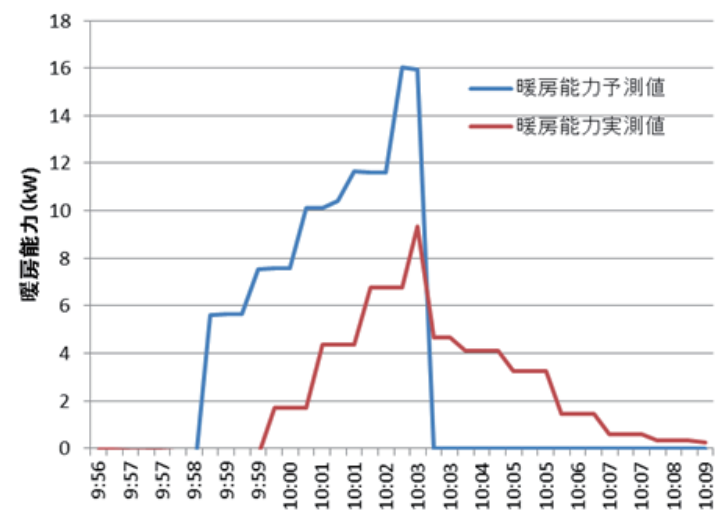


図 182 発停に伴う空調能力挙動の実測・予測比較

図 182 からわかる様に、暖房運転時は室内機がサーモオンし、冷媒が流れてもすぐに吹き出し温度が上がらず、また運転後にサーモオフした場合でも、室内機に残った熱が室内に供給され続けるため、サーモオン時間中だけの比較評価では実際の挙動を模擬出来ない。

しかし、この結果を踏まえて、1 日の運転時間全体の能力を実測と予測で比較したところ、表 75 に示したように十分な精度で評価できていることがわかった。このことから、サーモオフ中に室内機が出力している能力はデータ上の機器挙動から把握することは出来ないため、今回作成した特性式を用いてサーモオン中に“空調機としては”能力を出しているとして計算することでシステムの消費電力を正しく評価できるものとする。

表 75 暖房能力予測の1日積算値評価

実測能力積算値	予測能力積算値	誤差
9.27kWh	10.46kWh	13%

よって暖房についても特性式としてはこれで完成とする。

3.3.2.4 システム予測の評価（冷房：時間平均データを使用した場合）

次に、3.3.1 で作成したビルマルチエアコン全体の計算モデルの妥当性を評価する。
まず、Aビルにおける1時間毎の平均データを用いた評価結果を以下に記す。

(1) ビルマルチエアコンの能力算出

ビルマルチエアコンの計算モデルは入力として空調負荷データを必要とする。そこでまず運転データから能力を算出する必要がある。

室外機の空調能力をコンプレッサカーブ法（CC法）を用いて算出した。

CC法は室内機入口・出口間の冷媒エンタルピー差と圧縮機特性から算出された冷媒循環量を積算して室外機能力を求めるものである。

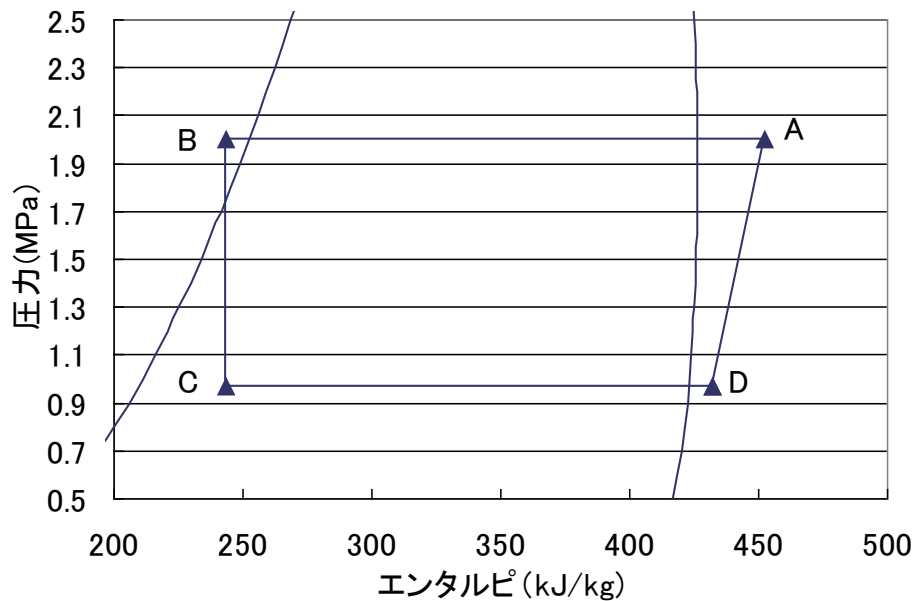


図 183 冷凍サイクルの冷媒状態遷移図（モリエル線図）

図 183 のモリエル線図で、点 C・D 間が冷房時室内機における冷媒の比エンタルピー変化を表している。よって、C・D 間の比エンタルピー差と冷媒循環量の積を算出することで室外機の冷房能力を算出することが可能となる。

室外機で把握しているサーモオンしている室内機容量と、上記 CC 法によって算出された空調能力を比較した。

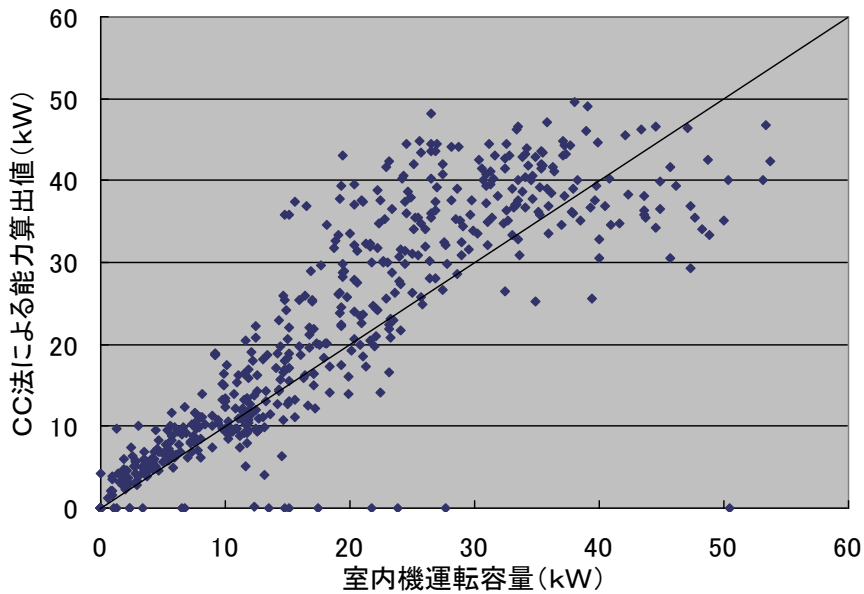


図 184 CC法による空調能力評価結果（冷房）

実際にすべての室内機的能力計測を行った上での比較ではなく、サーモオン中室内機の定格能力の積算値に過ぎないため、ある程度ばらついた結果になっている。全体的な傾向としては、サーモオンしている室内機が全数定格能力を出力していたとしても CC 法にて算出した空調能力の方が大き目になっている。これは室内機特性式の精度評価でも述べたように、発停しているデータの平均値をもとにしている影響であると考えられる。すなわち、CC 法による計算では、運転データを平均することによって、実際の稼働時よりも低圧を高め想定してしまい、圧縮機吸入冷媒密度が実際よりも高く算出され、結果的に同じ圧縮機回転数でも冷媒循環量を過大に算出してしまうことで、実際よりも高い空調能力を予測している。

しかしながら、全体的に高めの値を算出しつつも、定性的には空調負荷の変動をある程度予測出来ていると考えられる。室外機特性式精度評価に際しては、CC 法による能力を正として、消費電力予測精度の確認を行う。

(2) ビルマルチエアコンシステム消費電力予測精度評価

シミュレーションを用いて、前項で算出した空調能力と計測された室内・室外の温湿度条件をもとに電力消費量を予測し、実測値と比較を行った。

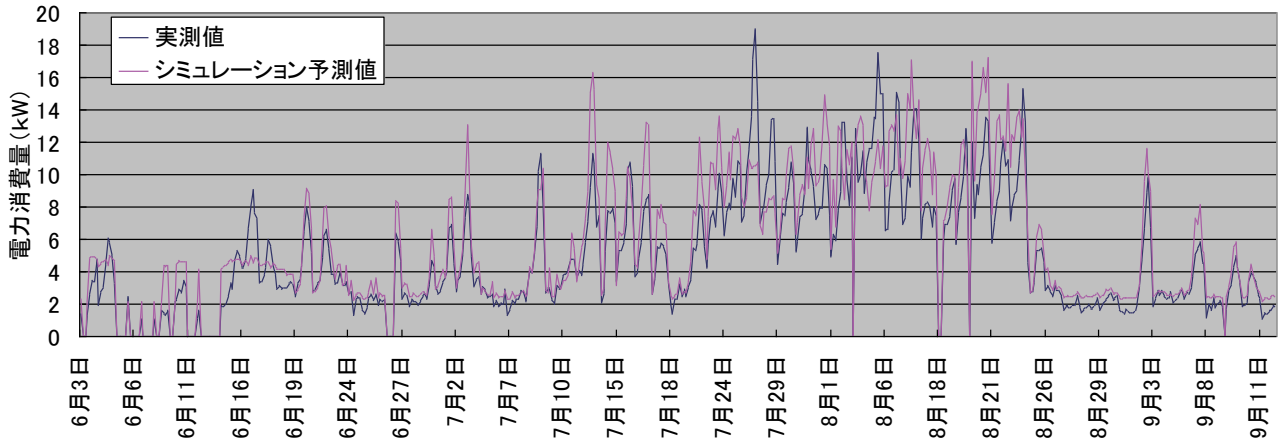


図 185 電力消費量予測値・実測値比較

部分毎に比較すると乖離もあるが、全体のトレンドとしては電力消費量の変動をうまく予測出来ているものと考ええる。

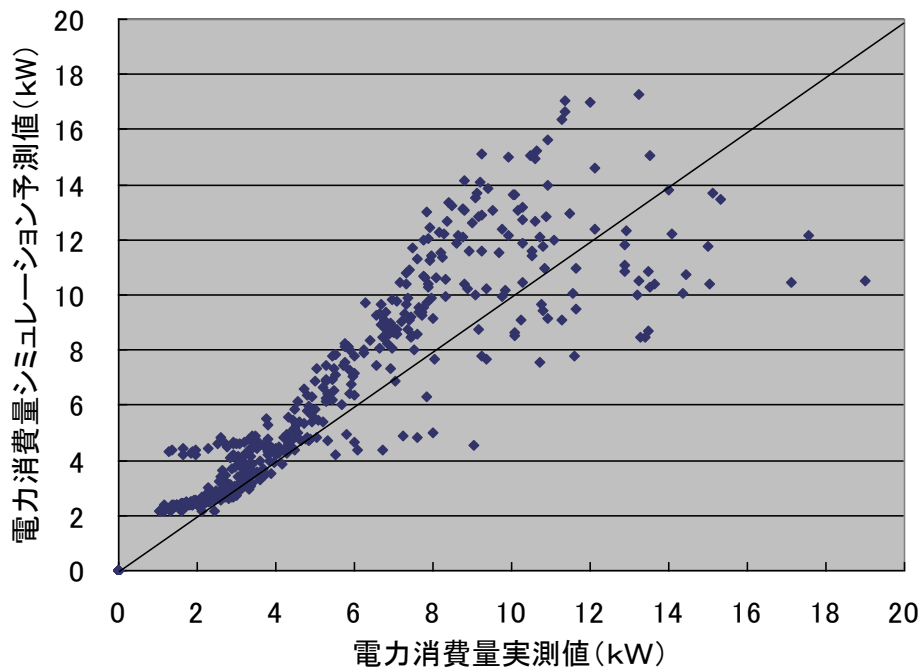


図 186 電力消費量予測精度検証

図 186 のように実測値と予測値の比較を散布図で示すと、全体の傾向として消費電力を過大に予測していることが分かる。これは図 184 で示したように、空調負荷の予測値が上振れしており、その負荷を満たすようにシミュレーションを行った結果、消費電力も過大に予測されたものと考えられる。

また、予測値の方が大きいという全体の傾向とは別に、予測値よりも実測値の方が大きいデータが少数ながら含まれている。データを詳細に確認した結果、シミュレーション上はすべての室内機が均等に運転しているとして計算しているが、実測結果では必ず室内機の何台かが停止しており、その結果稼働中の室内機の負荷率が上昇し、運転時の低圧が低下することにより消費電力が増大してしまっている。尚、夏季全体では、シミュレーションによる消費電力予測値は実測値よりも約 16%大きかった。

3.3.2.5 システム予測の評価（暖房：時間平均データを使用した場合）

冷房の場合と同じく、Aビルにおける暖房時のシステム予測の妥当性を評価する。

(1) ビルマルチエアコンの能力算出

暖房の場合は冷房の場合とは逆に、図 183 の点A-B間が暖房時室内機における冷媒の比エンタルピー変化を表している。よって、A-B間の比エンタルピー差と冷媒循環量の積を算出することで室外機の暖房能力を算出することが可能となる。

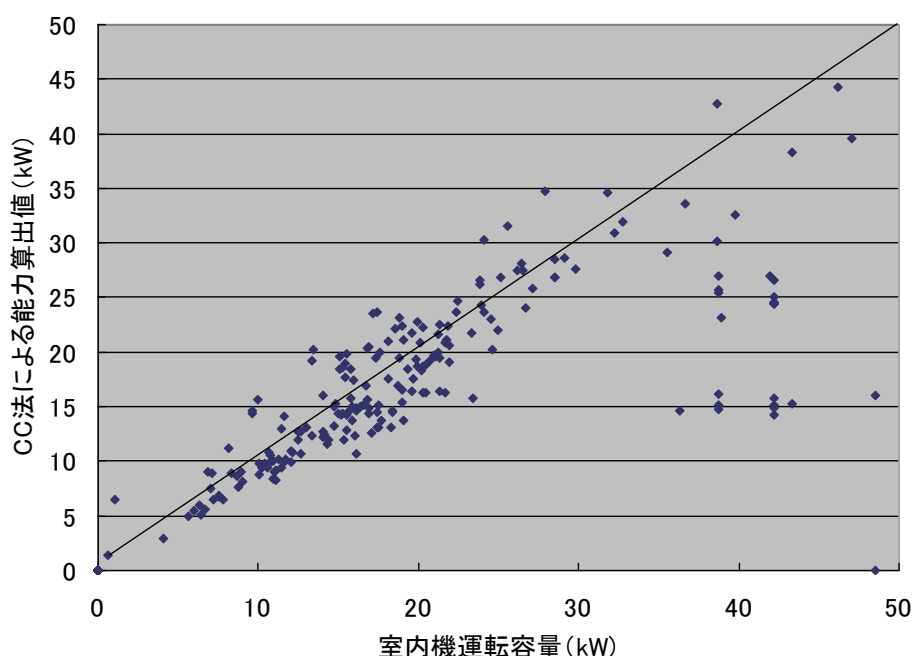


図 187 CC法による空調能力評価結果（暖房）

図 187 から、暖房の場合も能力はCC法から大まかに予測可能であることが確認された。ここで図 184 と図 187 を比較して、暖房時の方がデータのバラつきが小さいのは、Aビルに導入したビル用マルチ空調機がヒートポンプデシカント調湿外調機との連携に配慮した高頭熱仕様となっており、冷房時の蒸発温度を負荷に合わせて変更する機能を有しているがゆえに、単純にサーモオンしている室内機容量からでは空調能力が推定しづらいことが要因と考えられる。

室外機特性式精度評価に際しては、CC法によって算出された空調能力を正として、消費電力予測精度の確認を行う。

(2) ビルマルチエアコンシステム消費電力予測精度評価

シミュレーションを用いて、CC法で算出した空調能力と計測された室内・室外の温湿度条件をもとに電力消費量を予測し実測値と比較を行った。

ただし、今回評価したビル用マルチ空調機は冷房・暖房同時運転が可能な機種であり、冬季データの中にも同時運転を行っているデータが含まれていたが、冷暖同時運転時を模擬するためのシミュレーションは未完成であるため、ここでは全室内機が暖房設定になっているデータのみを用いて評価を行っている。

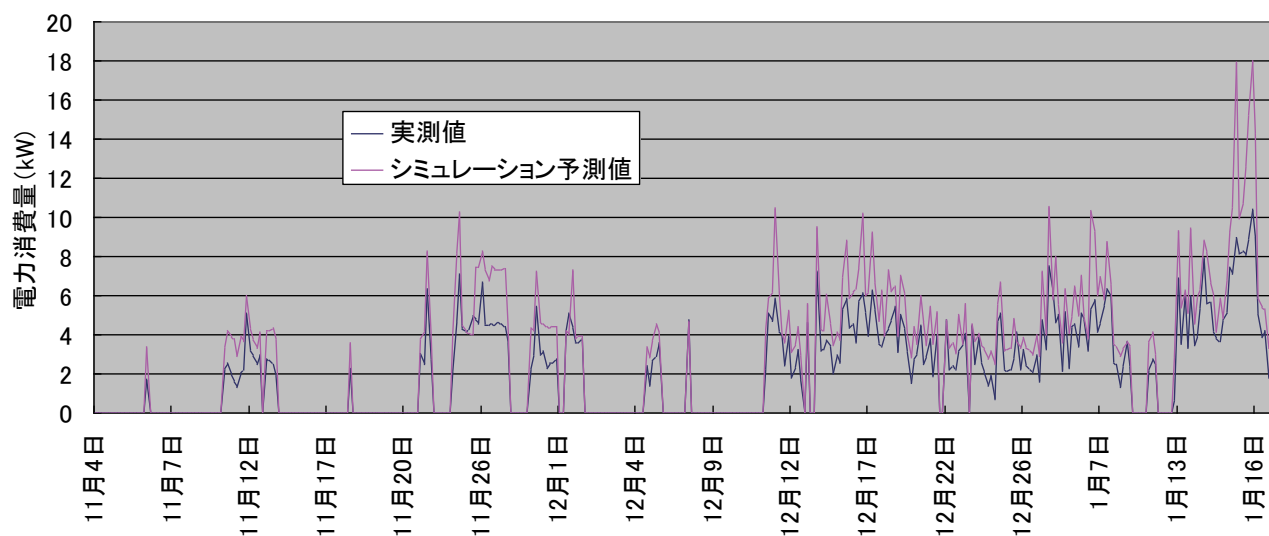


図 188 電力消費量予測値・実測値比較

部分毎に比較すると乖離もあるが、全体のトレンドとしては電力消費量の変動をうまく予測出来ている。

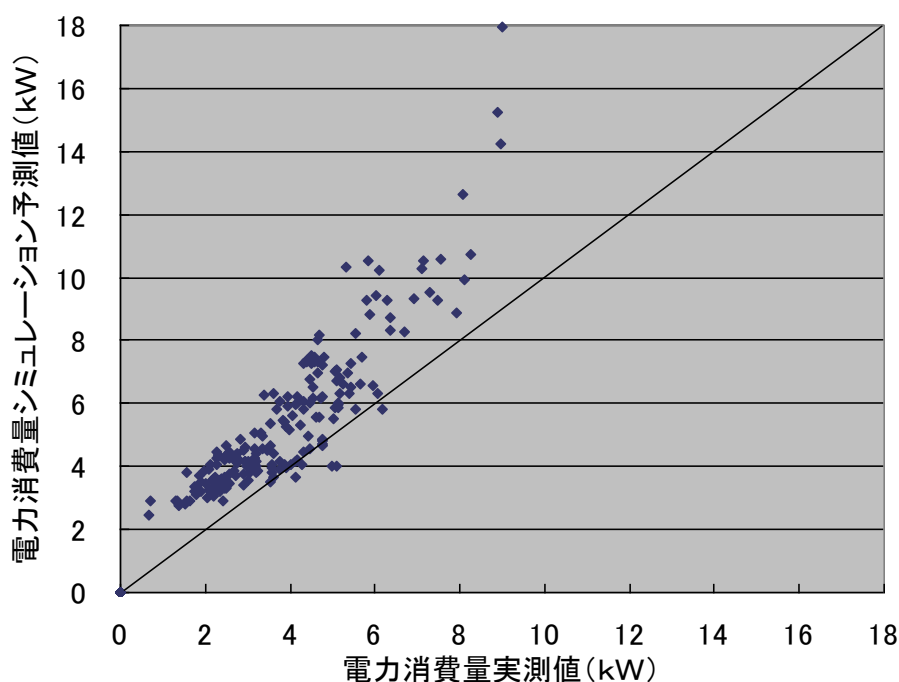


図 189 電力消費量予測精度検証

図 189 のように実測値と予測値の比較を散布図で示すと、全体の傾向として消費電力を過大に予測していることが分かる。特に低負荷（低消費電力）領域での乖離の原因は、暖房運転時にはサーモオフした室内機はファン風量を低下させるため、ファン動力が大幅に低下するが、シミュレーション上では室内機個別の発停は無いものとして計算しているため、その効果が反映されていないことが原因だと考えられる。

逆に、高負荷（高消費電力）領域での乖離の原因は、当該運転時の負荷率が高く且つ外気温度も低いことにより、低圧が低下しており、シミュレーションに用いた圧縮機特性データ作成時の運転条件からは外挿になっているため、予測精度が低いものと考えられる。

尚、冬季全体では、シミュレーションによる消費電力予測値は実測値よりも約 41%大きかった。上記のようにいくつかの課題はあるが、図 188 からわかるように、定性的なビル用マルチの挙動自体はかなり高い精度で再現出来ている。

3.3.2.6 システム予測の評価（1分毎の瞬時データを使用した場合）

図 186、図 189 で、システム消費電力の予測値が実測値よりも、特に高負荷側で大きい傾向が見られた。値に誤差はあるものの、その誤差にはある一定の傾向があるため、特性式の形自体ではなく、特性パラメータに要因があると考えられる。前項までのシミュレーションに用いていた室外熱交のデータは、計測に用いている機種に近い機種の試験データをもとに作成したものであったが、上記の課題に鑑み、計測に用いている機種自体の試験データを新たに入手し、特性パラメータを新たに作成した。

Aビルに用いている室外機は、12HP機と16HP機の外マルチ接続による28HP機である。性能評価試験時の計測データを元に作成した外熱交のパラメータは表 76 のようになる。

表 76 Aビルに導入した室外機の熱交パラメータ

	冷房		暖房	
	12HP機	16HP機	12HP機	16HP機
B F	0.383	0	0.586	0.551
A	0.0289	0	0	1.85
B	0	0.297	0	0
C	0	2.39	0	0

図 184 に見られるような、一時間平均データを用いて予測したことによる空調負荷計算のばらつきも排して特性式を評価するため、1分毎の瞬時データを用いて試算を行った。

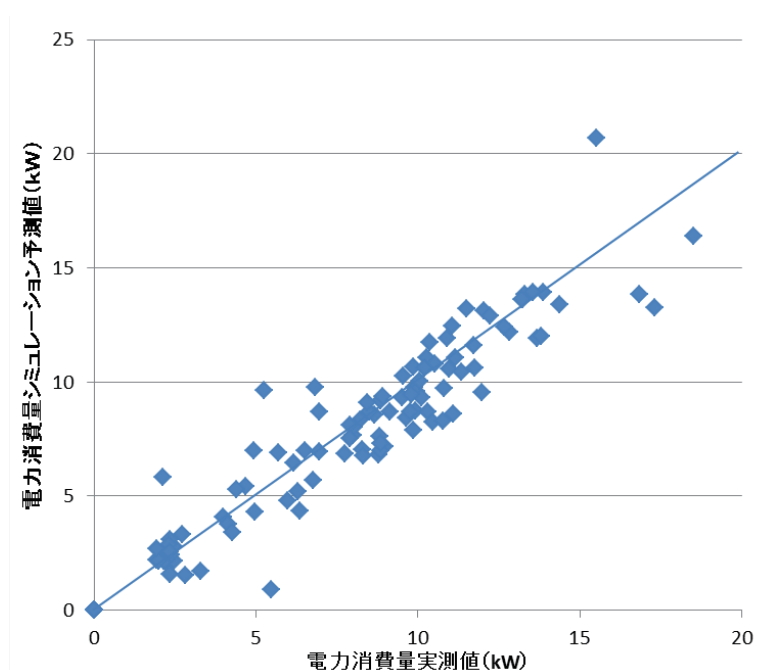


図 190 電力消費量予測精度検証（冷房：瞬時データ）

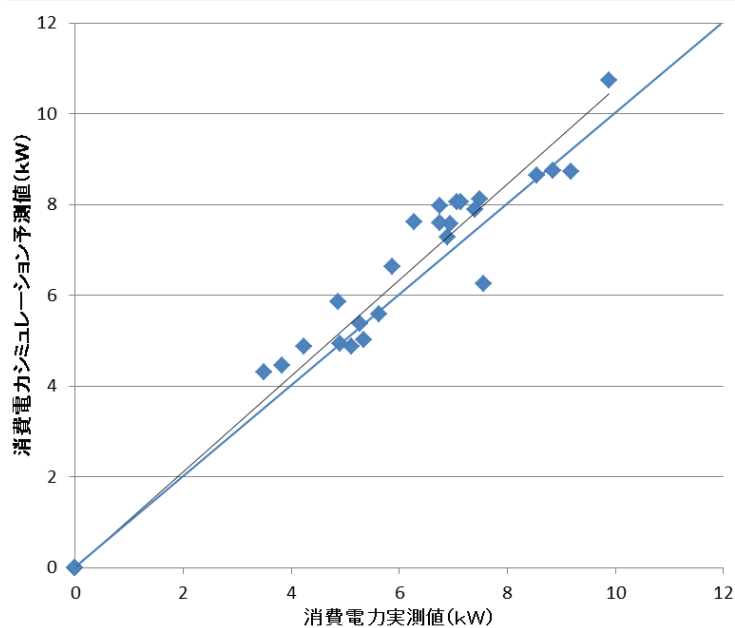


図 191 電力消費量予測精度検証（暖房：瞬時データ）

瞬時データを元に算出した結果を、単純に表示するとかなりばらつきのあるデータとなるため、図 190、図 191 においてはそれぞれ 10 分平均、1 時間平均の値を表示している（暖房時の方が前述のとおり発停の多い運転である為、平均時間を長くとらないと有意なデータにならなかった）。

以上より、瞬時データと適切な特性パラメータを使用することで、今回作成した計算モデルはかなり正確に消費電力を予測可能であることが確認された。

3.4 全熱交換機の熱交換能力特性の把握

3.4.1 実測による全熱交換機の熱交換能力特性の把握

顕熱潜熱交換効率評価

実際に据え付けられた全熱交換器の顕熱回収能力、潜熱回収能力を測定したが、それぞれで異なる能力を記した。また、カタログ性能と乖離した測定結果となった。これは、本体据付条件の影響で機外静圧が個々で異なり、その影響で内部漏れが発生し、有効換気量率が変化する事で能力がそれぞれで異なると思われる。図 192 図 193、表 77 に平成 26 年度にて報告した交換効率資料を記す。

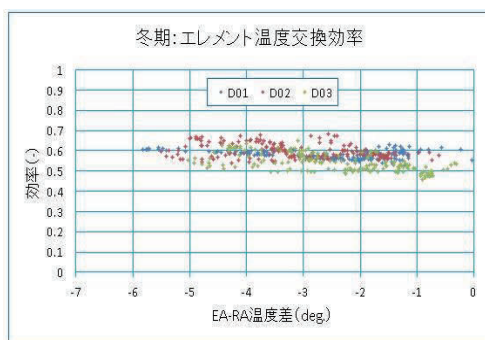


図 192 大阪 C ビル冬期エレメント温度交換効率

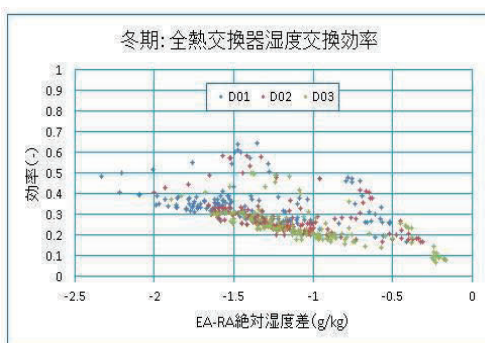


図 193 大阪 C ビル冬期エレメント絶対湿度交換効率

得られた結果より、効率を算出した。称呼値と実測値比較を以下の表に記す。D01 から D03 で能力にばらつきがみられる。

表 77 効率一覧

		称呼		実測(大阪 C)冬季のみ		
		夏季	冬季	D01	D02	D03
給気	乾球温度[°C]	30.0	18.2	15.7	16.0	15.2
	絶対湿度[g/kg・DA]	12.0	6.39	6.50	6.19	6.17
エレメント温度交換効率[%]		76.9		58.0	60.0	54.3
温度交換効率[%]		62.0	86.0	67.2	69.2	63.5
湿度交換効率[%]		61.8	31.0	34.6	25.0	24.4
エンタルピー交換効率[%]		62.0	65.0	54.7	52.3	48.5

3.4.2 全熱交換機内部漏れ原因

上記結果より、OA 機外静圧(ダクトの長さ、ダクトの曲がり、ダクトの径、外気取り込みグリル等)より RA 機外静圧(ダクトの長さ、ダクトの曲がり、ダクトの径、還気取り込みグリル等)が小さい場合は、主に機械内部で排気系統から給気系統へ空気が漏れると考えられる。ただし、外気側と排気側に至っても、圧力差が存在した場合は、同様に給気系統から排気系統に空気が漏れると考えられる。図 194 に漏れイメージを記す。内部漏れのパターンは 2 種類存在する。RA 機外静圧 < OA 機外静圧の場合と RA 機外静圧 > OA 機外静圧の場合である。RA 機外静圧 < OA 機外静圧の場合は、主に排気が給気に漏れて、RA 機外静圧 > OA 機外静圧の場合は、主に外気が排気に漏れると考えられる。図 195 に換気装置機械内部の圧力推移イメージを記す。機械入口段階で既に負圧となっており、フィルター、エレメント等によって更に内部が負圧となる。その際に、反対側の換気経路と圧力差が生じた場合に空気が漏れると考えられる。

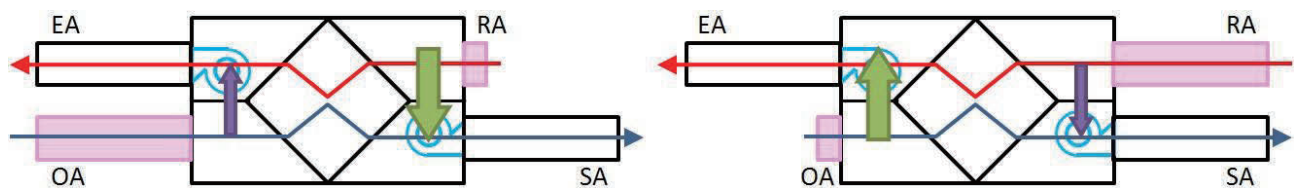


図 194 全熱交換器内部漏れイメージ図

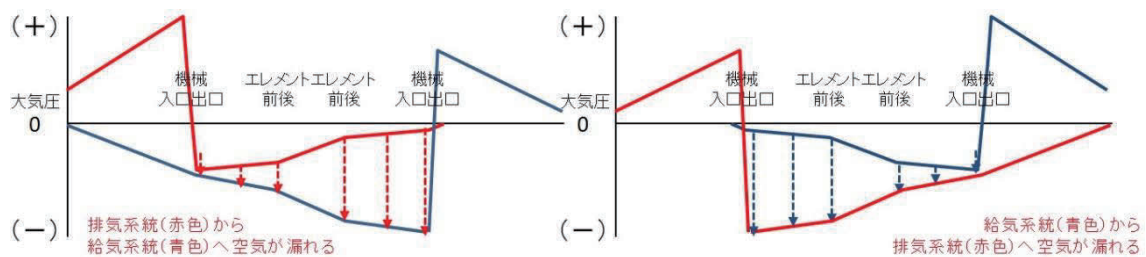


図 195 全熱交換器内部圧力推移イメージ図

上記の事より、機械内部の漏れが圧力差によって発生していると考えられ、その内部漏れによって能力が可変すると考えられる。よって、実際の据付状態を調査した。

3.4.2.1 実際のダクトの設置状況

図 196、図 197 に東京 B ビルと大阪 C ビルの図面を記す。それぞれにて全く異なるダクト経路となっており、ビル毎に取り回しが異なる事が良くわかる。また、東京 B ビルは建屋縦シャフトに外気取入れ用ダクトが据え付けられているので、計算を行うことが出来なかった為、大阪 C ビルの OA、RA の機外静圧を『国土交通省：建築設備設計基準』に準じて算出した。表 78 計算によって求めた機外静圧一覧に記す。

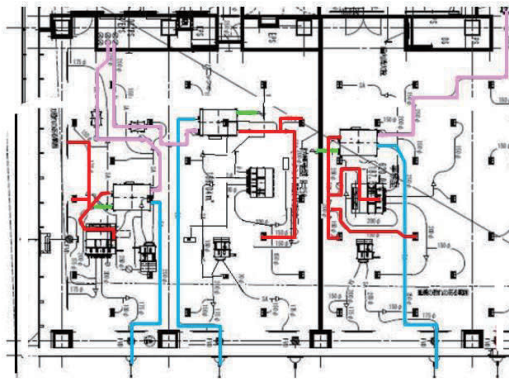


図 196 東京 B ビル（赤色給気、青色排気、桃色外気）

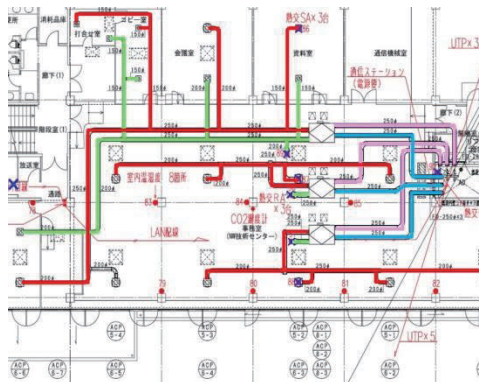


図 197 大阪 C ビル（赤色給気、青色排気、桃色外気）

表 78 計算によって求めた機外静圧一覧

	OA 経路[Pa]	RA 経路[Pa]
D01	80	99
D02	85	9
D03	84	9

表では D02 と D03 では数値上の変化がないが、結果としては異なる結果が出ている。これは、図面と実際に施工する際の差（曲げや縦方向へダクト施工等）が影響していると考えられる。

上記の事も踏まえ、内部漏れが原因で顕熱回収効率、潜熱回収効率に個体差が発生する事もわかった。しかし、実際に据え付けられた状態の内部漏れ量について測定した知見がないため、東京 B ビルにて実際に有効換気量の測定を実施した。実際の能力を記すため、平成 26 年度報告の東京 B ビルの顕熱回収効率、潜熱回収効率を図 198、図 199 に記す。

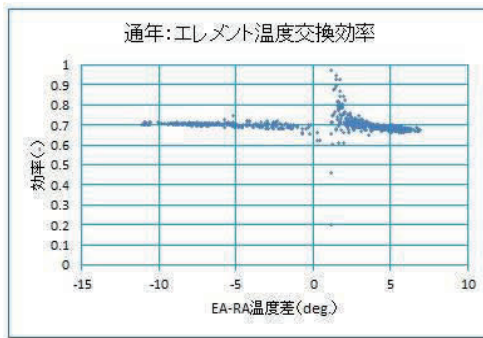


図 198 東京 B ビル通年エレメント温度交換効率

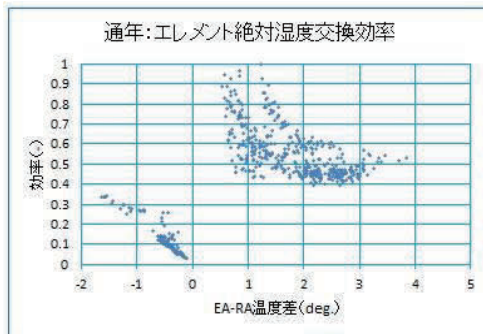


図 199 東京 B ビル通年エレメント絶対湿度交換効率

得られた結果より、効率を算出した。称呼値と実測値比較を以下の表 79 に記す。カタログ性能と比較して顕熱回収効率は概ね達成していたが、冬期における潜熱回収効率は未達となっていた。カタログ値 31.0%に対して、実測値 8.7%である。本事象となった原因を明らかにするために、有効換気量率を測定した。

表 79 称呼値と実測結果

		称呼		実測(東京 B)	
		夏季	冬季	夏季	冬季
給気	乾球温度[°C]	30.0	18.2	30.6	17.3
	絶対湿度[g/kg・DA]	12.0	6.39	12.4	5.67
エレメント温度交換効率[%]		76.9		70.0	
温度交換効率[%]		62.0	86.0	55.1	79.2
湿度交換効率[%]		61.8	31.0	49.3	8.7
エンタルピー交換効率[%]		62.0	65.0	52.0	52.2

3.4.3 東京 B ビル有効換気量試験

有効換気量試験については、JIS B 8628 全熱交換器と JRA4056 全熱交換器有効換気量試験方法に準拠して試験を実施した。表 80 に計測装置の仕様一覧と、試験内容と結果を 3.4.3.1～3.4.3.6 に記す。

表 80 計測装置仕様表

メーカー名	型番	仕様
TSI	Model8371	測定範囲 風量：42-4,250m ³ /h 差圧：±3,735Pa 圧力 256-1,016mmHg 湿度：0-95%RH 温度：-40℃～121℃ 精度 風量：読み値の±3% 差圧：読み値の±3% 圧力：読み値の±2% 湿度：±3%RH 温度：±0.3℃(0～71℃) ±1.2℃(-40～0℃,71～121℃) 分解能 風量：1m ³ /h 差圧：0.001Pa 圧力：1mmHg 湿度：0.1%RH 温度：0.1℃
コーナー札幌	KNS-二酸化炭素	測定方法：非分散型赤外線吸収方式 (NDIR) ポンプ吸引式 測定範囲：0～5000ppm 分解能：1ppm 精度：±50ppm、または測定値の5% (25℃において)

3.4.3.1 現状据え付けられている状態の機外静圧測定、SA 風量測定

全熱交換器の内部漏れは設置条件により異なる為、吹出し口、吸込み口の機外静圧が内部漏れに影響を与える。よって、現状の据付状態の機外静圧を測定する。具体的には、吹出し口、吸込み口に測定用ダクトを設置し、微差圧計にて機外静圧を測定する。また、有効換気量を測定する為に、SA 吹出し口に風量測定装置を設置し、風量を実測する。尚、対象の全熱交換器は SA 吹出し口が 4 ヶ所だったので、全ての箇所にて測定を実施した。図 200 に設置イメージ図、図 201 に機外静圧測定結果(時系列)、表 61 に機外静圧測定結果、図 202 に SA 口の図面、写真 16 SA 風量測定様子に風量測定時の写真、表 82 に風量測定結果を記す。



図 200 微差圧計設置、風量計測

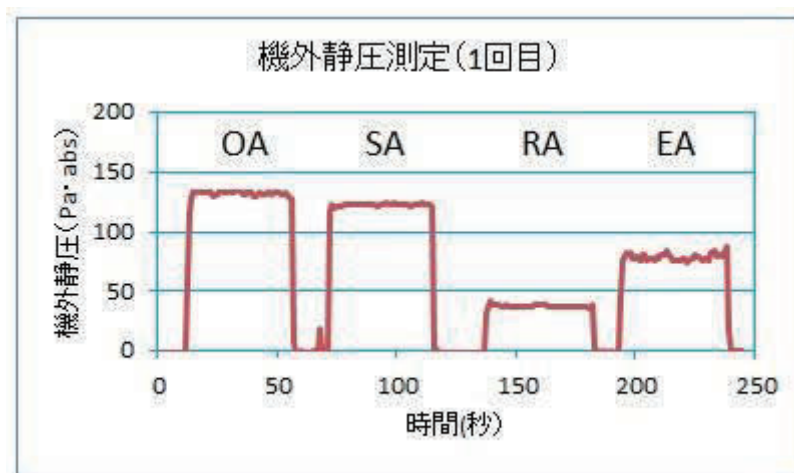


図 201 機外静圧測定 (時系列)

表 81 機外静圧測定結果

機外静圧	OA	SA	RA	EA
1 回目 [Pa]	133.1	122.8	38.4	79.4
2 回目 [Pa]	132.5	121.9	38.1	80.7
平均 [Pa]	132.8	122.3	38.2	80.0

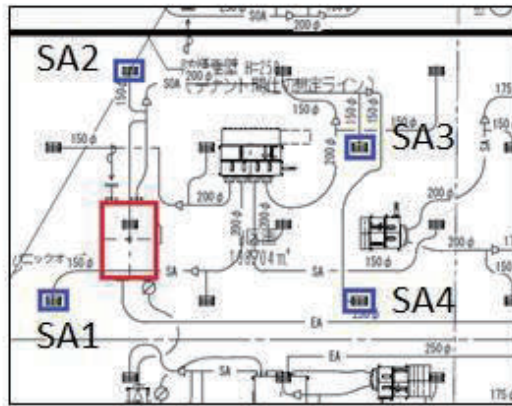


図 202 図面 SA 口吹出し位置



写真 16 SA 風量測定様子

表 82 風量測定結果

風量	SA1	SA2	SA3	SA4	合計
1 回目 [m ³ /h]	388	21 ^(※)	21 ^(※)	260	690
2 回目 [m ³ /h]	398	21 ^(※)	21 ^(※)	256	696
平均値 [m ³ /h]					693

※ 風量系表示部はゼロを表示したが、手で感じる程度の風は明らかに出ているので、測定レンジの下限値以下だと考えられる。計算上では、最低測定レンジの 5 割[21 m³/h]とした。

3.4.3.2 二酸化炭素濃度測定のために、ダクト、補助ファン取付

天井リターン方式にて換気装置が据付けられている為、効果的に二酸化炭素を散布する事が出来ない。よって、RA 測定用ダクトにフレキシブルダクトと、静圧調整用の補助ファンを取り付ける事で、十分に攪拌した空気を機械内部へ送る。より攪拌する為に、補助ファンの上流に二 U 酸化炭素封入口を設置する。図 203 設置イメージ図、写真 17、写真 18 に RA 吸込み口ダクト接続写真、写真 19 に補助ファン設置写真を記す。

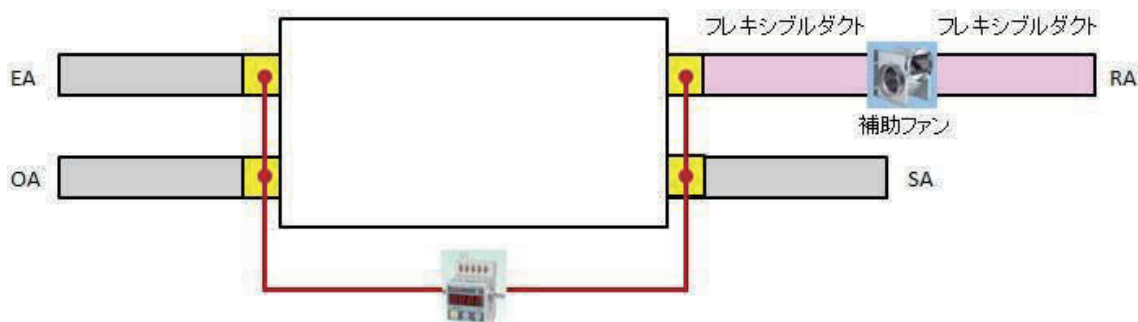


図 203 設置イメージ図



写真 17 既存 RA 吸込み口



写真 18 RA 吸込み口に補助ダクトとフレキシブルダクトを追加



写真 19 補助ファン設置

3.4.3.3 機外静圧調整作業

既設の全熱交換器にフレキシブルダクト、補助ファンを取り付けたため、実際に据え付けられていた状態から異なる据付状況になっている。よって、補助ファンを任意に可変運転出来る為にインバータ装置を取り付けて、ファン回転数を調整し、実際の据付状態の機外静圧を達成する。図 204 に設置イメージ図、表 83 に調整後の機外静圧を記す。概ね、元々の据付条件の機外静圧を満足することが出来た。

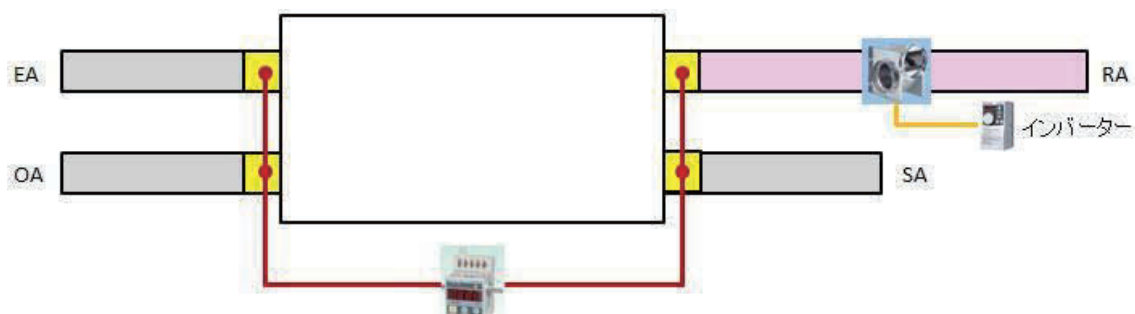


図 204 INV を用いて機外静圧調整

表 83 機外静圧調整結果

機外静圧	OA	SA	RA	EA
元々の据付条件値[Pa]	132.8	122.3	38.2	80.0
補助ファンにて据付条件再現結果[Pa]	133.9	124.5	36.7	81.5
差[Pa]	1.1	2.2	▲1.5	1.5

3.4.3.4 RA 風量測定

インバータ装置による機外静圧調整作業にて静圧を整えたので、実際の据付条件と同等条件になっている。全熱交換器は風量比によっても、性能が異なる。そこで、対象吸込み口にて風量を測定する。図 205 に設置イメージ図、写真 20 に RA 吸込み口、表 84 に風量測定結果を記す。

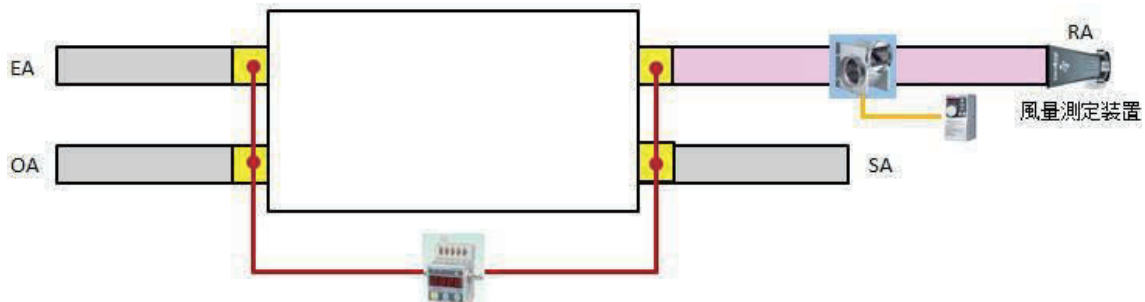


図 205 設置イメージ図



写真 20 RA 風量測定

表 84 RA 風量測定結果

	1 回目	2 回目	3 回目	平均値
RA 風量[m ³ /h]	795	785	790	790

3.4.3.5 二酸化炭素散布、二酸化炭素濃度サンプリング実施

有効換気量を測定する状況が整ったので、実測を行う。尚、JIS B 8628 全熱交換器と JRA4056 全熱交換器有効換気量試験方法に準拠して、RA 二酸化炭素濃度を 5000ppm 近傍に調整し、SA と OA の二酸化炭素濃度を測定し、二酸化炭素移行率を測定し、漏えい量、有効換気量、有効換気量率を求める。図 206 に設置イメージ図、計算式、図 207、図 208 にサンプリング結果(時系列)、表 85 にサンプリング結果、表 86 に有効換気量結果を記す。

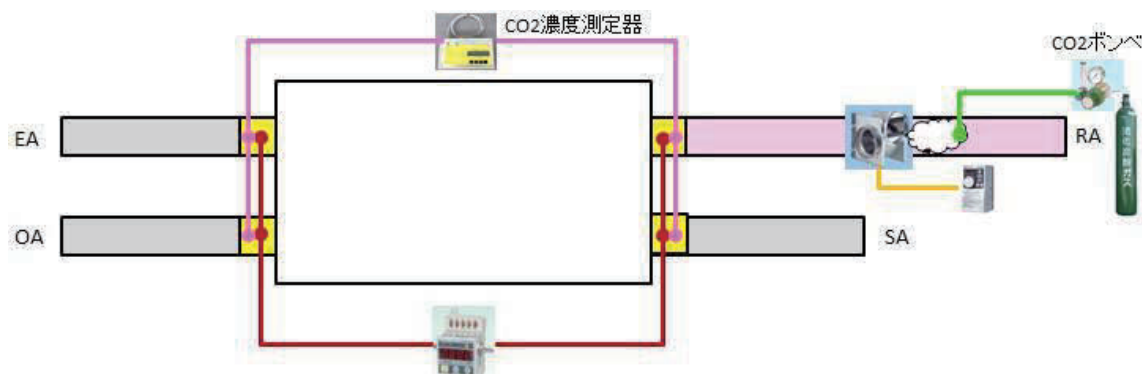


図 206 二酸化炭素散布、二酸化炭素濃度サンプリング実施

計算式を下記に記す。

a. 二酸化炭素移行率

$$E_0 = \frac{C_{SA} - C_{OA}}{C_{RA} - C_{OA}} \times 100$$

式 132

b. 漏えい量

$$q = Q_s \times \frac{E_0}{100}$$

式 133

c. 有効換気量

$$Q_E = Q_s - q$$

式 134

d. 有効換気量率

$$E = 100 - E_0$$

式 135

E_0	: 二酸化炭素移行率	[%]
C_{SA}	: 給気二酸化炭素濃度	[ppm]
C_{OA}	: 外気二酸化炭素濃度	[ppm]
C_{RA}	: 還気二酸化炭素濃度	[ppm]
q	: 漏洩量	[m ³ /h]
Q_s	: 給気量	[m ³ /h]
Q_E	: 有効換気量	[m ³ /h]
E	: 有効換気量率	[%]

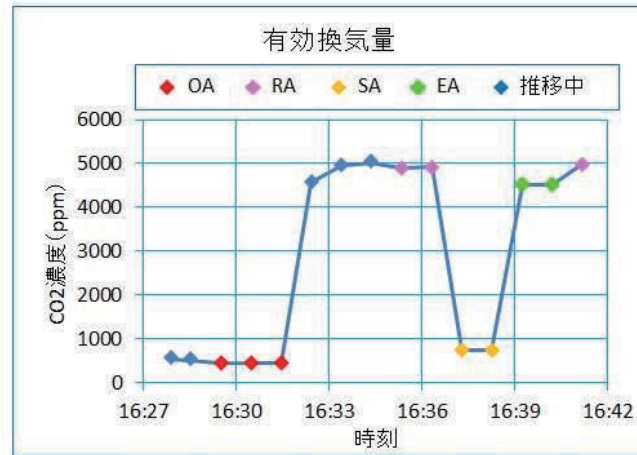


図 207 CO₂ 濃度サンプリング（時系列）

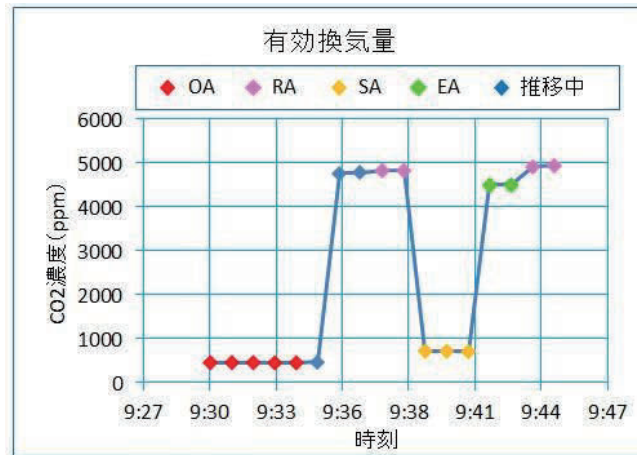


図 208 CO₂ 濃度サンプリング（時系列）

表 85 サンプリング結果

二酸化炭素濃度	OA	RA	SA	EA
1 回目 [ppm]	456	4908	747	4527
2 回目 [ppm]	452	4832	712	4502
平均値 [ppm]	454	4870	730	4515

表 86 有効換気量率

有効換気量	E_0	q	Q_E	E
1 回目	6.5[%]	45.0[m ³ /h]	648.0[m ³ /h]	93.5[%]
2 回目	5.9[%]	40.9[m ³ /h]	652.1[m ³ /h]	94.1[%]
平均値	6.2[%]	43.0[m ³ /h]	650.0[m ³ /h]	93.8[%]

3.4.3.6 内部漏れ量を考慮した冬期湿度交換効率

有効換気量率は 93.8% となった。本結果を踏まえて、冬期に実測した湿度交換効率 8.7% と同等であるかを検討する為に、実際の計測値を図 209 に記す。

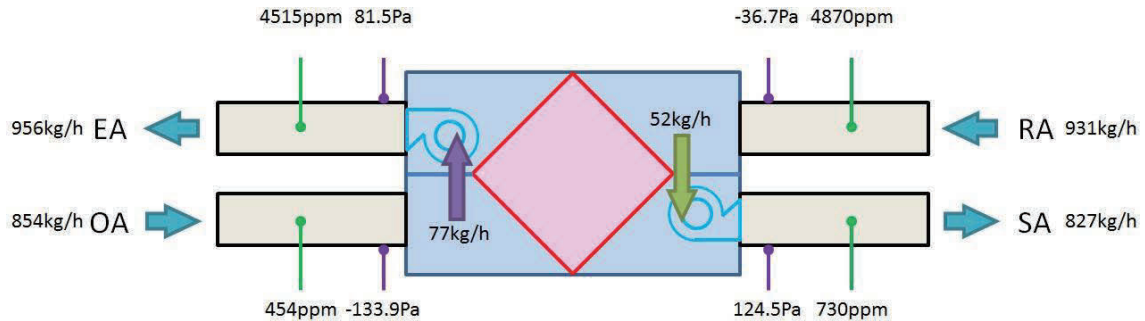


図 209 イメージ図に実測データ記載

昨年度の測定結果(昨年度報告内容と同等の効率時の性能(一部抜粋))を表 87 に記す。

表 87 昨年度測定結果

実測値	温度[°C]				効率		絶対湿度[g/kg]				効率
	OA	SA	RA	EA	$\frac{EA - RA}{OA - RA}$		OA	SA	RA	EA	$\frac{EA - RA}{OA - RA}$
測定日											
2/5 11:00	10.3	16.4	24.5	15.6	62.6		1.43	7.16	6.28	5.84	9.1

エレメントにおける絶対湿度交換効率は 9.1% であり、SA 側にどれだけ漏れているかを検討する。

検討① 水分量の整理

給気する水分量 = 外気湿度流入分 + エレメントによる回収分 + 内部漏れ湿度

式 136

$$\text{水分計算式} = \text{絶対湿度} \times \text{風量} \times \text{密度} \quad \dots \quad \left[\frac{g}{h} \right] = \left[\frac{g}{kg} \times \frac{m^3}{h} \times \frac{kg}{m^3} \right]$$

式 137

$$\text{外気流入分} = \text{外気絶対湿度} 1.43 \times \text{質量流量} 777 = 1111.1 \left[\frac{g}{h} \right]$$

式 138

$$\text{内部漏れ} = \text{還気絶対湿度} 6.28 \times \text{質量流量} 51.6 = 324.2 \left[\frac{g}{h} \right]$$

式 139

エレメント移動分 = (還気絶対湿度 - 外気絶対湿度) × 実測湿度回収効率

$$\text{エレメント移動分} = (\text{還気絶対湿度 } 6.28 - \text{外気絶対湿度 } 1.43) \times 0.091 = 0.44 \left[\frac{g}{h} \right]$$

式 140

$$\text{エレメント湿度回収} = \text{エレメント移動絶対湿度 } 0.44 \times \text{質量流量 } 879 = 386.8 \left[\frac{g}{h} \right]$$

式 141

$$\text{給気空気絶対湿度} = (1111.1 + 324.2 + 386.8) \div 827 = 2.20 \left[\frac{g}{h} \right]$$

式 142

内部漏れを見込んだ給気側湿度交換効率

$$\eta = \frac{X_{OA} - X_{SA}}{X_{OA} - X_{RA}} \times 100 = \frac{1.43 - 2.20}{1.43 - 6.28} \times 100 = 15.9 [\%]$$

式 143

内部漏れを含んだ絶対湿度交換効率は 15.9% となった。

次に、カタログ条件の測定方法と、本測定方法について比較を実施した。測定機種のカatalogを抜粋し、測定方法を赤線にて記す。

機種名		VKMP50HM					
電源		単相200V					
換気モード		全熱交換換気			普通換気		
ノッチ		特強	強	弱	特強	強	弱
外気負荷	冷房時★1 (kW)	4.83 (2.03)					
処理能力	暖房時★2 (kW)	5.76 (2.56)					
	電動機出力 (kW)	0.280×2					
送風機	風量 (m³/h)	500/500	500/500	400/390	500/500	500/500	400/390
	機外静圧 (Pa)	225/225	170/140	125/85	225/225	170/140	125/85
フレッシュアップ運転	給気過剰	—	1.08:1/1.11:1	1.22:1/1.28:1	—	1.08:1/1.11:1	1.22:1/1.28:1
	風量比率 (給気:排気)	—	1:1.08/1:1.11	1:1.22/1:1.28	—	1:1.08/1:1.11	1:1.22/1:1.28
加湿器	加湿方式	流下式自然蒸発加湿器					
	加湿量 (kg/h) ★3	2.9/2.9 (4.2/4.2)		—	—		
	給水圧力 (MPa)	0.02~0.49					
消費電力 (W)		560/655	505/570	415/440	560/655	505/570	415/440
温度交換効率 (%) ★4		76/76	76/76	77/77.5	—		
エンタルピー	冷房時★1	68/68	68/68	71/71.5	—		
	交換効率 (%) ★4	72/72	72/72	74/74.5	—		
運転音 ★5 (dB)		37.5/37.5	36/33	33/29	38.5/37.5	36/33	33/29
製品質量 (満水時) (kg)		102 (105)					

★1.空気条件:室内側27°CDB・19°CWB、室外側35°CDBによる値です。 ★2.空気条件:室内側20°CDB、室外側7°CDB・6°CWBによる値です。 ★3.空気条件:室内側20°CDB・15°CWB、室外側7°CDB・6°CWBによる値です。加湿量 < > 内の数値は全熱交による湿度回収分を含んだ値です。 ★4.効率測定にあたっては、定格の機外静圧を室外側と室内側の比を7対1の割合でつけた状態で測定しています。なお、温度交換効率は冷房時、暖房時の平均値を示します。 ★5.運転音はJIS B 8628に準拠した無響室換算値です。実際に据え付けた状態で測定すると、周囲の騒音や反射を受け表示値より大きくなるのが普通です。 ※外気負荷処理能力、加湿量は、特強および強ノッチ定格風量時のものです。数値内()は、全熱交による熱回収分を内数で示しています。 ※電流、消費電力、効率は上記風量時の値です。 ※温度交換効率、エンタルピー交換効率、加湿量は、給気と排気の風量比や空気条件により変動します。 ※空気条件は一般居室の場合であり、温度差の大きな冷蔵庫等では、表記数値内であっても使用できません。 ※除霜運転中も換気量を維持するため、送風機は運転を続けます。吹出口からのドラフトが懸念される場合は、吹出方向に注意してください。 ※加湿関連の注意事項、空気条件による加湿量の算出、交換効率、熱交換器(直膨コイル)能力の算出、運転音、使用周囲条件の詳細は、技術編P.182をご覧ください。
※ビル用マルチ室外ユニットとの接続容量を計算する時には、3.5kWとして計算してください。

図 210 カタログ抜粋

赤線抜粋：「効率測定にあたっては、定格の機外静圧を室外側と室内側の比を 7 対 1 の割合でつけた状態で測定しています。なお、温度交換効率は冷房時、暖房時の平均値を示します。」カタログ値と同等条件にするために、実測値の静圧比を算出した。現状の比は 3.65 対 1 となっており、差圧は 97.2Pa、漏れ量は 51.6kg/h である。本条件を模擬的に 7 対 1 に変換した場合、静圧比 1.54 倍となり漏れ量は 77.7kg/h となり、カタログ条件の空気条件と、実際の温湿度条件によって、交換量は可変し(20°C DB,15°CDB⇒10.3°CDB,2.7°CWB)、風量比によっても能力は変更するので本条件では能力 1.06 倍となり、諸条件を考慮すると能力は 26.1%の効率となる。また、エレメント表面に付着する微細粉末の影響によっても性能は低下すると考えられる。カタログ値 31.0%に対して、少し乖離があるが、経年劣化の影響も加味されて、31.0%から少し回収効率が低下して 26.1%程度の効率になったのではないかと考えられる。よって、実測した冬期の絶対湿度回収効率 8.7%は妥当であると考えられる。

上記有効換気量率試験によって、内部漏れの影響がある事が良くわかった。

3.4.4 国立研究開発法人 建築研究所 環境実験棟 有効換気量試験

3.4.4.1 設置環境、試験条件

東京 B ビルでの有効換気量率試験によって、内部漏れの影響が有効換気量率に大きく寄与する事が判明したので、機外静圧を任意に変更した場合の漏れ影響を国立研究開発法人建築研究所環境実験棟にて評価した。

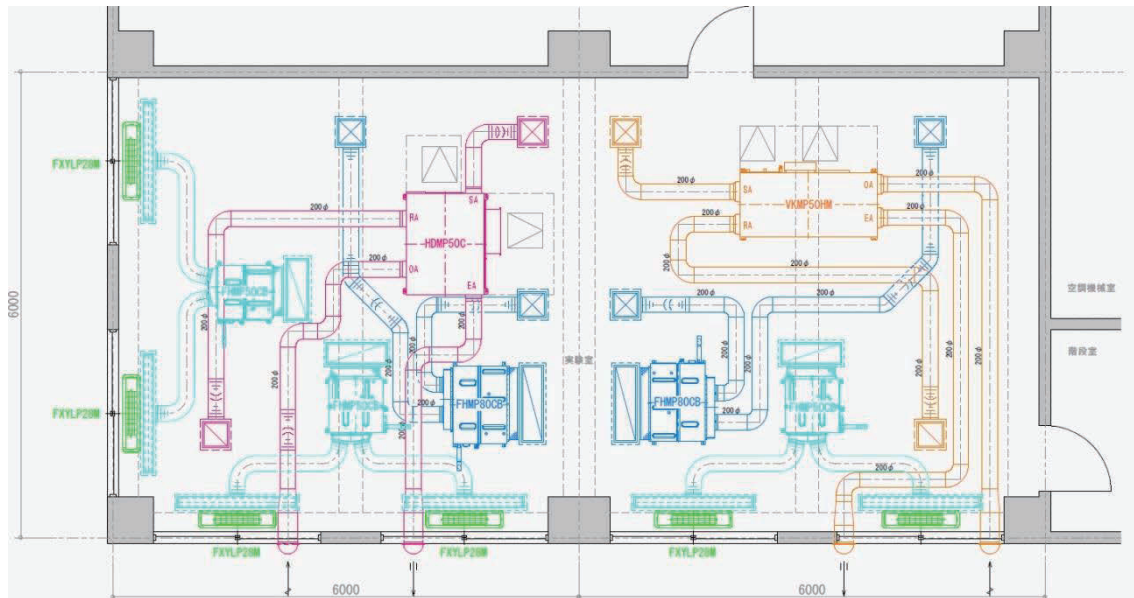


図 211 全熱交換器据付位置

PAC1-G 外気処理機(全熱交換器加湿内蔵タイプ)VKMP50HM にて機外静圧を任意に変更し、有効換気量を測定する。しかし、対象の室は模擬オフィスであり、定常的にダクトを取回すことが出来ない為、室内側の RA 口、SA 口に取り外し可能なダクトシステムを取り付けた。外気と繋がる OA 口と EA 口に関しては、外壁に同等のシステムを据付することが出来ない為、ダクト経路に追加で補助ファンと操作する為のインバータを取り付けた。尚、既存の RA 口、SA 口に取り付ける為、ボックスを加工してフレキシブルダクトと接続した。写真 21、写真 22 に記す。



写真 21 RA 側補助ファン、補助ダクト取付位置



写真 22 SA 側補助ファン、補助ダクト取付位置

各種様々な計測器と補助ファンを取り付けたので、まとめたイメージを図 212 に記す。

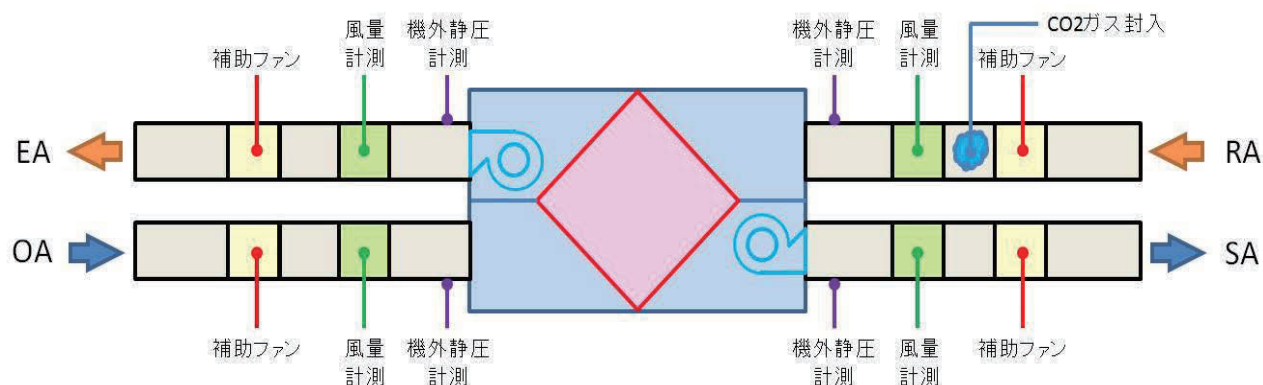


図 212 全熱交換器試験時、測定内容まとめ

内部漏れ量を評価する為に表 88 の試験条件に設定した。

表 88 試験条件一覧

No.	機外静圧[Pa]		機外静圧差[Pa]	风量[m ³ /h]	
	OA	RA	RA - OA	SA	EA
1	112.5	172.5	- 60	500	500
2	112.5	142.5	- 30	500	500
3	112.5	112.5	0	500	500
4	112.5	82.5	30	500	500
5	112.5	52.5	60	500	500
6	145	145	0	400	400

試験を実施する事で、排気経路から給気経路へ漏れる量と、給気経路から排気経路へ漏れる量が明確になり、风量を変えた場合の影響も評価する事が出来る。

3.4.4.2 試験結果

(1) 測定値一覧

表 89 測定結果

条件	機外静圧[Pa]		機外静圧差[Pa]	風量[m ³ /h]	
	OA	RA	OA - RA	SA	EA
1	123	49	-74	485	502
2	115	82	-33	497	496
3	106	112	6	502	497
4	108	141	33	504	507
5	113	176	63	501	506
6	140	150	10	394	404

表 90 測定結果より有効換気量率算出

条件	CO ₂ 濃度[ppm]			漏れ率 [%]	漏洩量 [kg/h]*	有効換気量 [kg/h]*	有効換気量率 [%]
	OA	SA	RA				
1	445	1479	4838	23.6	134	435	76.4
2	438	1108	4478	16.6	98	494	83.4
3	448	882	4596	10.5	62	531	89.5
4	445	774	4715	7.7	46	549	92.3
5	446	642	4535	4.8	29	565	95.2
6	444	854	4481	10.2	47	416	89.8

※ 本試験においては、模擬オフィスにて実験を実施している為、OA、SA、RA、EA それぞれの空調条件を同等にすることが出来ない為、質量流量にて評価を実施した。

(2) 時系列機外静圧、風量、質量流量

(l) <条件 1>

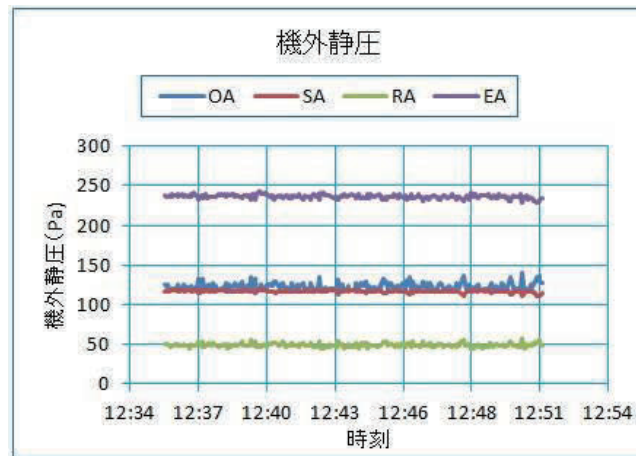


図 213 条件 1 : 機外静圧

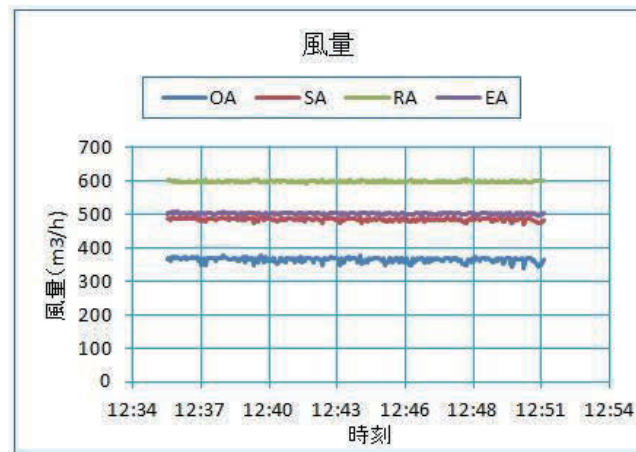


図 214 条件 1 : 風量

(II) <条件 2>

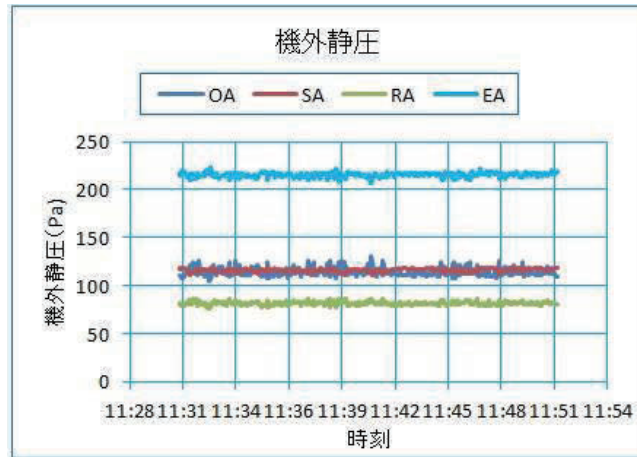


図 215 条件 2 : 機外静圧

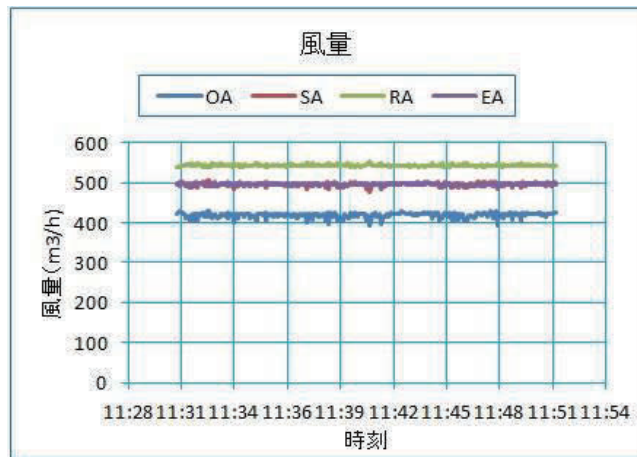


図 216 条件 2 : 風量

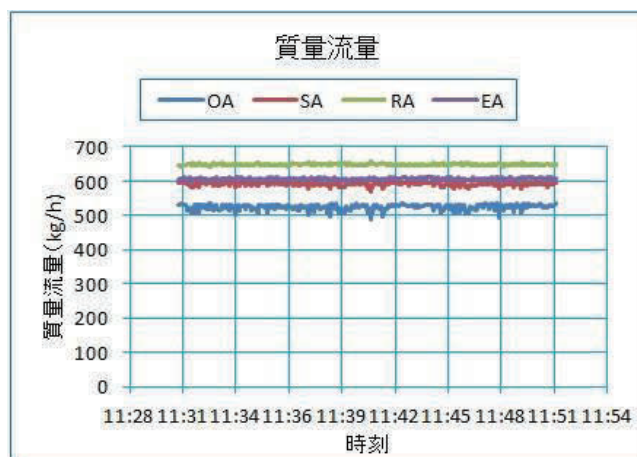


図 217 条件 2 : 質量流量

(III) <条件 3>

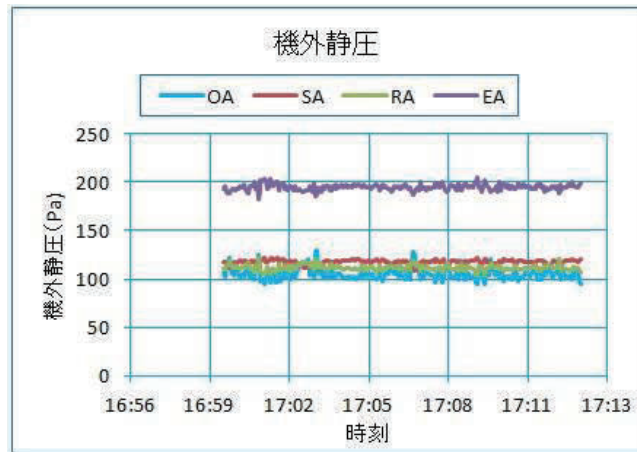


図 218 条件 3 : 機外静圧

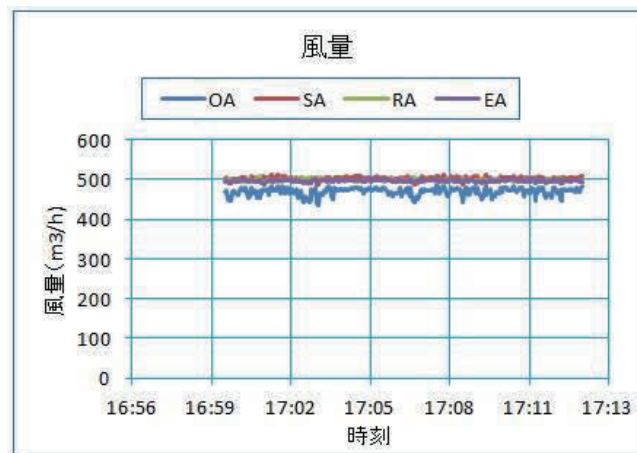


図 219 条件 3 : 風量

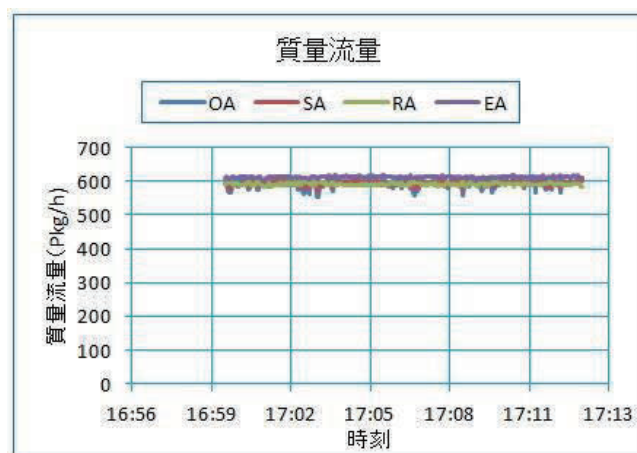


図 220 条件 3 : 質量流量

(IV) <条件 4>

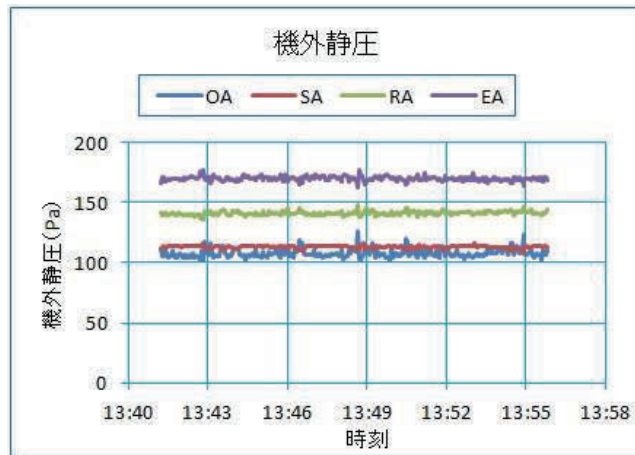


図 221 条件 4 : 機外静圧

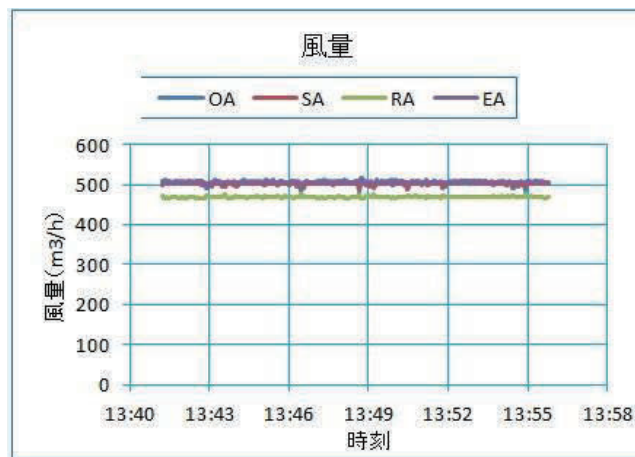


図 222 条件 4 : 風量

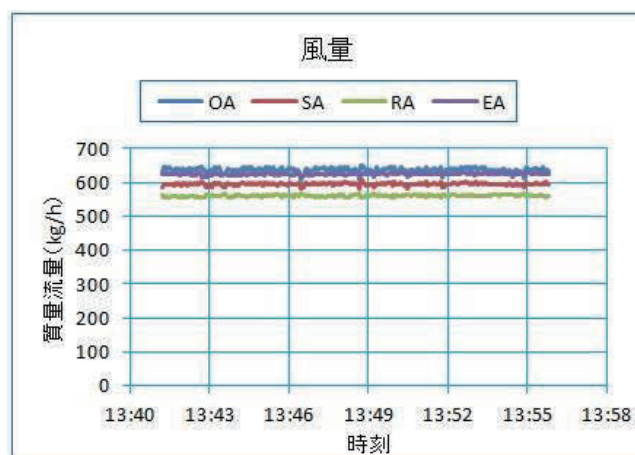


図 223 条件 4 : 質量流量

(V) <条件 5>

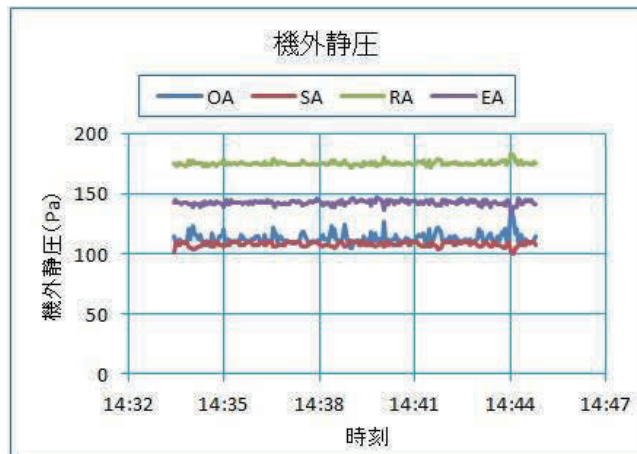


図 224 条件 5 : 機外静圧

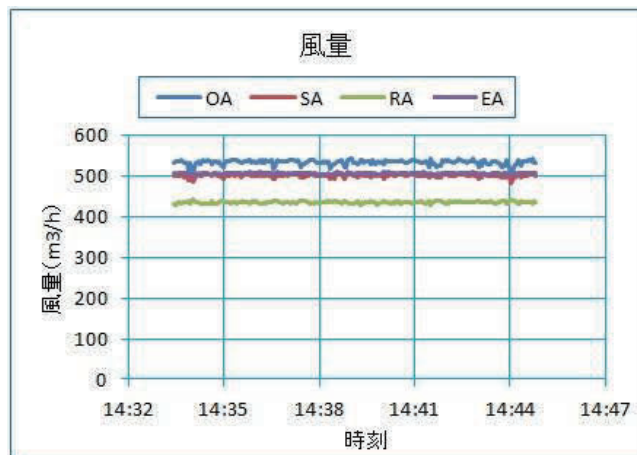


図 225 条件 5 : 風量

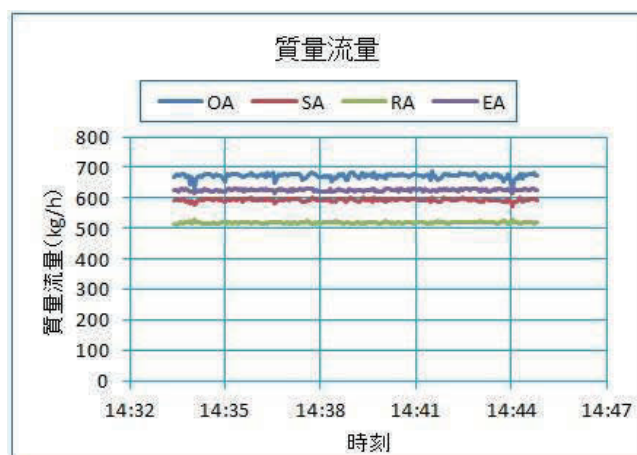


図 226 条件 5 : 質量流量

(VI) <条件 6>

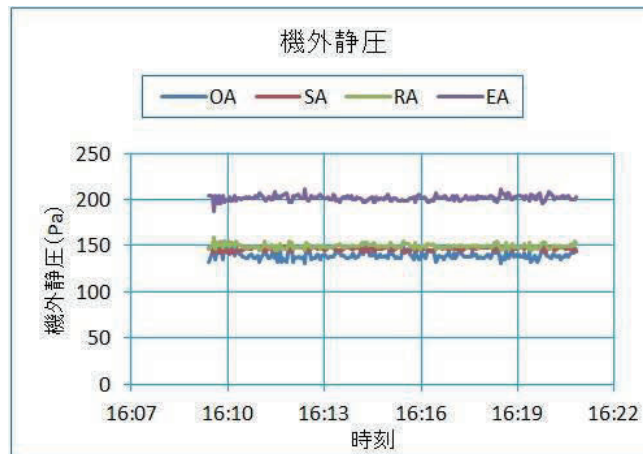


図 227 条件 6 : 機外静圧

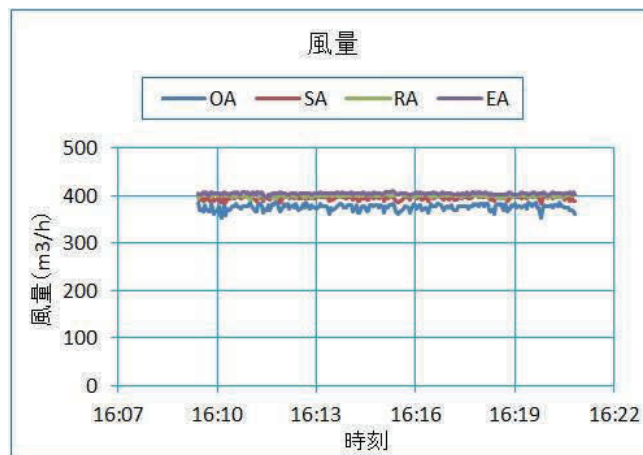


図 228 条件 6 : 風量

(3) 時系列 CO₂ 濃度

以下に測定時の CO₂ 濃度推移をそれぞれ示す。

(I) <条件 1>

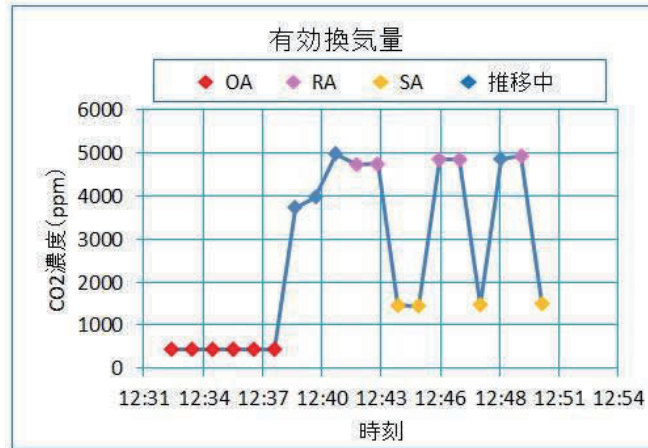


図 229 条件 1 : CO₂ 濃度の推移

(II) <条件 2>

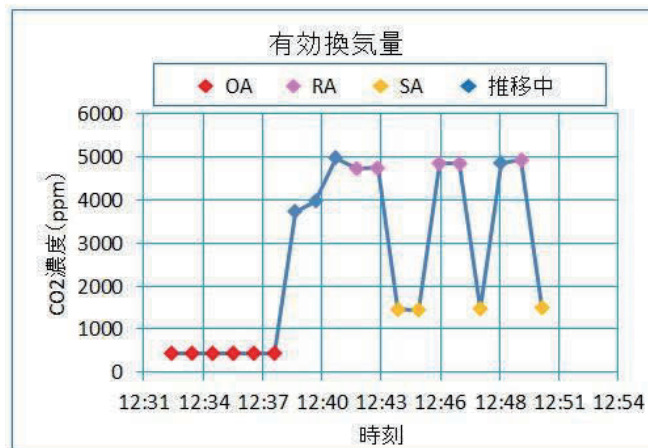


図 230 条件 2 : CO₂ 濃度の推移

(III) <条件 3>

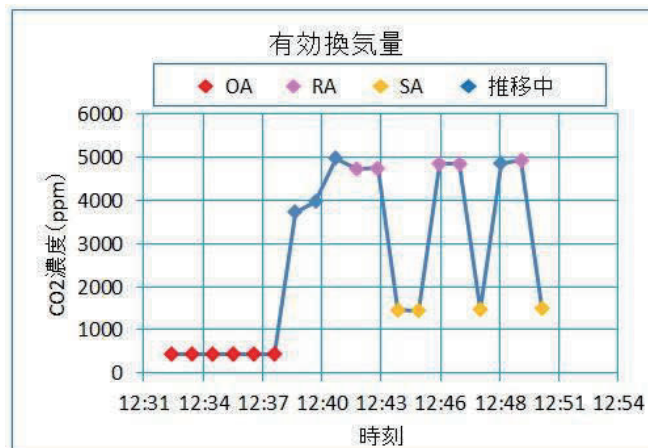


図 231 条件 3 : CO₂ 濃度の推移

(IV) <条件 4>

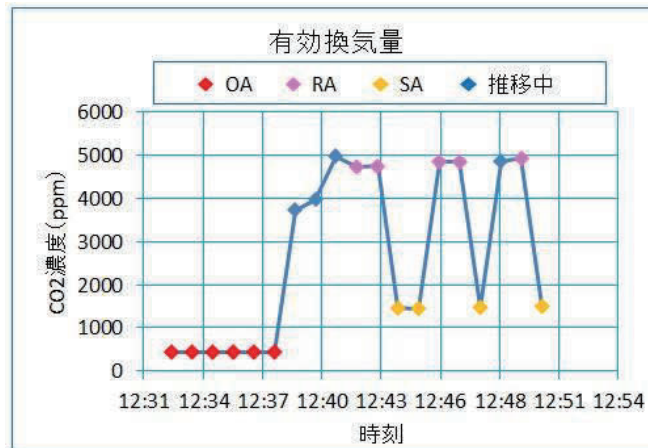


図 232 条件 4 : CO₂ 濃度の推移

(V) <条件 5>

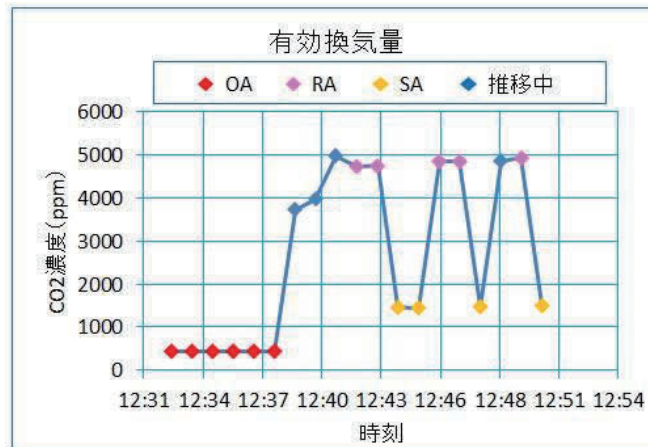


図 233 条件 5 : CO₂ 濃度の推移

(VI) <条件 6>

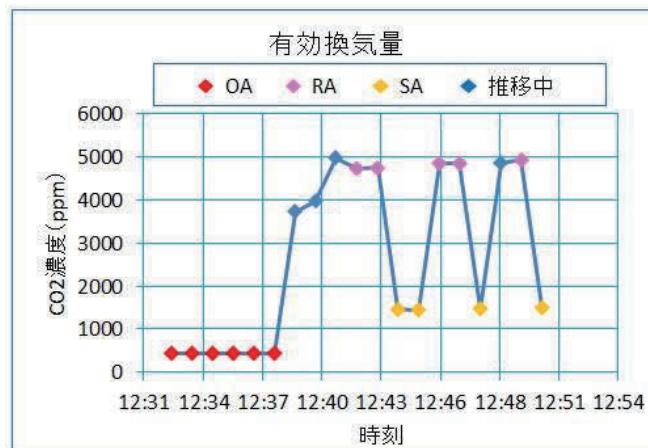


図 234 条件 6 : CO₂ 濃度の推移

(4) 機械内部漏れ量の把握

上記データより、風量が 500m³/h 時の吸込み口機外静圧差 RA-OA と有効換気量率を明確にした。以下にグラフを記す。想定した通り、吸込み口機外静圧差が RA<OA になるほど機械内部での漏れ量が増えた。

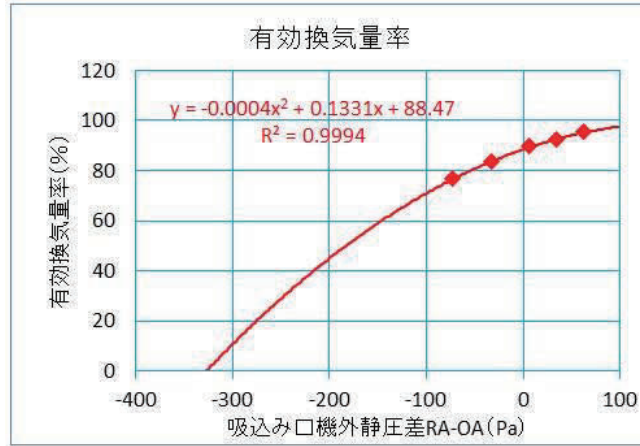


図 235 有効換気量率 (500 m³/h)

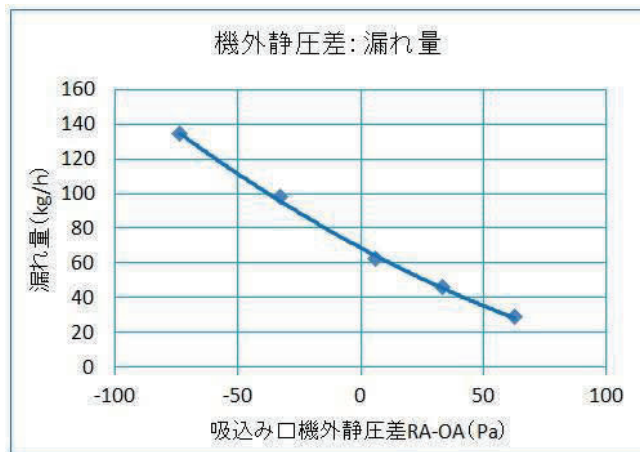


図 236 機外静圧と漏れ量の関係 (RA⇒SA 漏れ)

上記機外静圧と漏れ量のグラフより近似式を求めた。

RA⇒SA 漏れ量[kg/h] $y = 0.001824 x^2 - 0.760524 x + 68.831334$

式 144

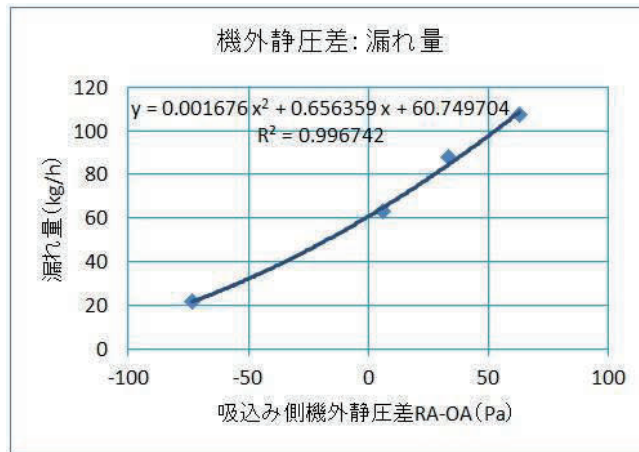


図 237 機外静圧と漏れ量の関係 (OA⇒EA 漏れ)

上記機外静圧と漏れ量のグラフより近似式を求めた。

$$\text{RA} \Rightarrow \text{SA 漏れ量}[\text{kg/h}] \quad y = 0.001676 x^2 - 0.656359 x + 60.749704$$

式 145

また、漏れ量は風量にかかわらず、吸込み側機外静圧差によって漏れる事から、風量が異なる場合の有効換気量率を求めた。以下に記す。風量が増えるほど、漏れ量の比率が低下する為、有効換気量率が高くなる傾向となった。したがって、風量が減れば減るほど、漏れ量の比率が増加する為、有効換気量率が低下する結果となった。

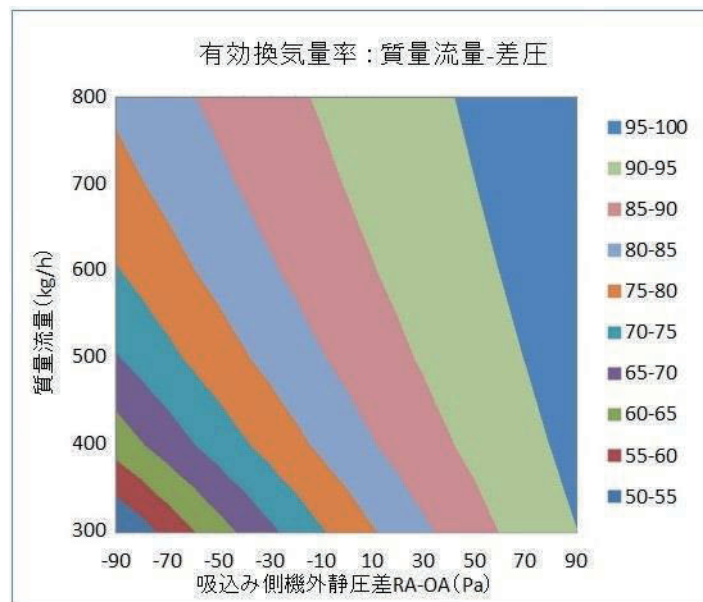


図 238 有効換気量率 (風量別)

(5) 風量比がもたらす温度交換能力変動の把握

本プログラムは Ver.05 (住宅・住戸の省エネルギー性能の判定プログラム Ver.01.15) を参照する。

Ver.05(住宅・住戸の省エネルギー性能の判定プログラム Ver.01.15)
2015.10

付録 A 熱交換型換気設備

熱交換型換気設備の補正温度交換効率 η'_t は、式(1)により表される。

$$\eta'_t = \eta_t \times C_{tot} \times C_{eff} \times C_{bal} \times C_{leak} \quad (1)$$

ここで、

- η'_t : 熱交換型換気設備の補正温度交換効率
- η_t : 熱交換型換気設備の温度交換効率 (%)
- C_{tot} : カタログ表記誤差による温度交換効率の補正係数
- C_{eff} : 有効換気量率による温度交換効率の補正係数
- C_{bal} : 給気と排気の比率による温度交換効率の補正係数
- C_{leak} : 排気過多時における住宅外皮経由の漏気による温度交換効率の補正係数

である。

A.1 熱交換型換気設備の温度交換効率

温度交換効率 η_t は、JIS B 8628「全熱交換器」に規定された温度交換効率とする。顕熱交換型換気設備の場合も、JIS B 8628「全熱交換器」に規定された試験方法及び計算方法に則って計算することとする。いずれの場合も、温度交換効率の測定空気条件は暖房時とする。また、温度交換効率の値は、100 分の 1 未満の端数を切り下げた小数第二位までの値とする(パーセントを単位とする場合は、小数未満の端数を切り下げた整数の値とする)。ただし、温度交換効率 η_t が0.4を下回る場合、又は、定格条件における給気風量が定格条件における還気風量の半分未満、若しくは 2 倍より大きい場合は、熱交換型換気設備による暖房負荷の削減効果を見込むことはできない。また、温度交換効率 η_t が0.95を上回る場合は、温度交換効率 η_t を0.95とする。

A.2 カタログ表記誤差による温度交換効率の補正係数

カタログ表記誤差による温度交換効率の補正係数 C_{tot} は、0.95とする。

A.3 有効換気量率による温度交換効率の補正係数

有効換気量率による温度交換効率の補正係数 C_{eff} は、式(2)により表される値とし、100 分の 1 未満の端数を切り下げた小数第二位までの値とする。有効換気量率による温度交換効率の補正係数 C_{eff} が 0 未満の値となる場合は、有効換気量率による温度交換効率の補正係数 C_{eff} は 0 に等しいとする。

$$C_{eff} = 1 - \frac{\left(\frac{1}{e} - 1\right)(1 - \eta_t)}{\eta_t} \quad (2)$$

ここで、

- e : 全熱換気設備の有効換気量率
- η_t : 熱交換型換気設備の温度交換効率

である。全熱換気設備の有効換気量率 e は、5 章「換気設備」に規定される値とする。

A.4 給気と排気の比率による温度交換効率の補正係数

給気と排気の比率による温度交換効率の補正係数 C_{bal} は0.90を用いるか、以下に示す方法に依ることができ。

給気と排気の比率による温度交換効率の補正係数 C_{bal} は、式(3)により表され、100 分の 1 未満の端数を切

り下げた小数第二位までの値とする。

$$C_{bal} = \frac{\eta_{t,d}}{\eta_t} \quad (3)$$

ここで、

$\eta_{t,d}$: 当該住戸における補正設計風量比での熱交換型換気設備の温度交換効率

η_t : 熱交換型換気設備の温度交換効率

である。

当該住戸における設計補正風量比での熱交換型換気設備の温度交換効率 $\eta_{t,d}$ は、式(4)により表される。

$$\eta_{t,d} = \begin{cases} \eta_d & (V_{d,RA} > V_{d,SA}) \\ \eta_d \times R'_{vnt,d} & (V_{d,RA} \leq V_{d,SA}) \end{cases} \quad (4)$$

ここで、

η_d : 当該住戸における補正設計風量比での熱通過有効度

$R'_{vnt,d}$: 当該住戸における補正設計風量比

$V_{d,SA}$: 当該住戸における設計給気風量 (m³/h)

$V_{d,RA}$: 当該住戸における設計運気風量 (m³/h)

である。

当該住戸における補正設計風量比での熱通過有効度 η_d は、直交流型熱交換器の場合は式(5a)、向流-直交流複合型熱交換器の場合は式(5b)により表される。

$$\eta_d = 1 - e^{\left[\frac{e^{-N_d^{0.75} \cdot R'_{vnt,d}} - 1}{N_d^{-0.22} \cdot R'_{vnt,d}} \right]} \quad (5a)$$

$$\eta_d = \frac{1 - e^{\left[-(1-R'_{vnt,d}) \left(1 + \frac{b}{l} \frac{\sin \alpha \cos \alpha}{0.0457143 N_d^2 + 0.0691429 N_d + 0.9954286} \right) N_d \right]}}{1 - R'_{vnt,d} \cdot e^{\left[-(1-R'_{vnt,d}) \left(1 + \frac{b}{l} \frac{\sin \alpha \cos \alpha}{0.0457143 N_d^2 + 0.0691429 N_d + 0.9954286} \right) N_d \right]}} \quad (5b)$$

ここで、

N_d : 当該住戸における設計風量比での伝熱単位数

b : 向流-直交流複合型熱交換器の向流部の幅(m)

l : 向流-直交流複合型熱交換器の向流部の長さ(m)

α : 向流-直交流複合型熱交換器の向流部と直交流部の接続角度(°)

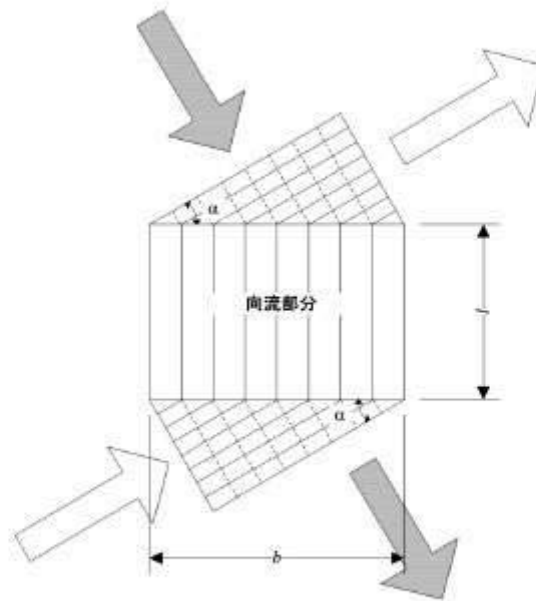


図 A.1 向流-直交流複合型熱交換器の幅、長さ、及び向流部と直交流部の接続角度である。ただし、向流-直交流複合型熱交換器の向流部の幅 b が0.01(m)以上2.0(m)以下、かつ向流部の長さ l が0.01(m)以上3.0(m)以下、かつ向流部と直交流部の接続角度 α が 15° 以上 60° 以下を満たしていない場合は熱交換機換気設備による暖房負荷の削減効果を見込むことはできない。

当該住戸における補正設計風量比 $R'_{vmt,d}$ は、式(6)により表される。

$$R'_{vmt,d} = \begin{cases} 1 - 10^{-8} & (\text{向流-直交流複合型かつ } R_{vmt,d} = 1) \\ R_{vmt,d} & (\text{それ以外}) \end{cases} \quad (6)$$

ここで、

$R_{vmt,d}$: 当該住戸における設計風量比

である。

当該住戸における設計風量比 $R_{vmt,d}$ は、式(7)により表される。

$$R_{vmt,d} = \begin{cases} \frac{V_{d,SA}}{V_{d,RA}} & (V_{d,RA} > V_{d,SA}) \\ \frac{V_{d,RA}}{V_{d,SA}} & (V_{d,RA} \leq V_{d,SA}) \end{cases} \quad (7)$$

ここで、

$V_{d,SA}$: 当該住戸における設計給気風量(m^3/h)

$V_{d,RA}$: 当該住戸における設計還気風量(m^3/h)

である。

当該住戸における設計風量比での伝熱単位数 N_d は、式(8)により表される。

$$N_d = N_{rtd} \times \frac{V_{rtd,min}}{V_{d,min}} \quad (8)$$

ここで、

- N_{rtd} : 定格条件における風量比での伝熱単位数
- $V_{rtd,min}$: 定格条件における最小風量 (m³/h)
- $V_{d,min}$: 当該住戸における設計最小風量 (m³/h)

である。

定格条件における最小風量 $V_{rtd,min}$ は、式(9)により表される。

$$V_{rtd,min} = \min(V_{rtd,SA}, V_{rtd,RA}) \quad (9)$$

当該住戸における設計最小風量 $V_{d,min}$ は、式(10)により表される。

$$V_{d,min} = \min(V_{d,SA}, V_{d,RA}) \quad (10)$$

式(11a)および式(11b)は、それぞれ直交流型熱交換器および向流-直交流型熱交換器における熱交換型換気設備の定格条件における補正風量比での熱通過有効度 η 、定格条件における補正風量比 $R'_{vnt,rtd}$ 及び定格条件における風量比での伝熱単位数 N_{rtd} の関係式である。この関係式を満たすように、熱交換型換気設備の定格条件における補正風量比での熱通過有効度 η 及び定格条件における補正風量比 $R'_{vnt,rtd}$ から定格条件における風量比での伝熱単位数 N_{rtd} を定めることとする。

$$\eta = 1 - e^{\left[\frac{e^{(-N_{rtd} \cdot 0.70 \cdot R'_{vnt,d})} - 1}{N_{rtd} \cdot 0.22 \cdot R'_{vnt,d}} \right]} \quad (11a)$$

$$\eta = \frac{1 - e^{\left[-1(1 - R'_{vnt,rtd}) \left(1 + \frac{\frac{b}{7} \sin \alpha \cos \alpha}{0.0457143N_{rtd}^2 + 0.0691429N_{rtd} + 0.9954286} \right) N_{rtd} \right]}}{1 - R'_{vnt,rtd} \cdot e^{\left[-1(1 - R'_{vnt,rtd}) \left(1 + \frac{\frac{b}{7} \sin \alpha \cos \alpha}{0.0457143N_{rtd}^2 + 0.0691429N_{rtd} + 0.9954286} \right) N_{rtd} \right]}} \quad (11b)$$

ここで、

- η : 定格条件における補正風量比での熱通過有効度
- $R'_{vnt,rtd}$: 定格条件における補正風量比

である。

定格条件における補正風量比での熱交換型換気設備の熱通過有効度 η は、式(12)により表される。

$$\eta = \begin{cases} \eta_t & (V_{rtd,RA} > V_{rtd,SA}) \\ \frac{\eta_t}{R'_{vnt,rtd}} & (V_{rtd,RA} \leq V_{rtd,SA}) \end{cases} \quad (12)$$

ここで、

- $V_{rtd,SA}$: 定格条件における給気風量 (m³/h)
- $V_{rtd,RA}$: 定格条件における還気風量 (m³/h)

である。

定格条件における補正風量比 $R'_{vnt,rttd}$ は、式(13)により表される。

$$R'_{vnt,rttd} = \begin{cases} 1 - 10^{-8} & (\text{向流-直交流複合型かつ } R_{vnt,rttd} = 1) \\ R_{vnt,rttd} & (\text{それ以外}) \end{cases} \quad (13)$$

定格条件における風量比 $R_{vnt,rttd}$ は、式(14)により表される。

$$R_{vnt,rttd} = \begin{cases} \frac{V_{rttd,SA}}{V_{rttd,RA}} & (V_{rttd,RA} > V_{rttd,SA}) \\ \frac{V_{rttd,RA}}{V_{rttd,SA}} & (V_{rttd,RA} \leq V_{rttd,SA}) \end{cases} \quad (14)$$

定格条件における給気風量 $V_{rttd,SA}$ 及び還気風量 $V_{rttd,RA}$ の代わりに、定格条件における外気風量 $V_{rttd,OA}$ 及び排気風量 $V_{rttd,EA}$ を用いることができる。

定格条件における給気風量 $V_{rttd,SA}$ 及び還気風量 $V_{rttd,RA}$ 、又は定格条件における外気風量 $V_{rttd,OA}$ 及び排気風量 $V_{rttd,EA}$ は、JIS B8628「全熱交換器」に定める値とする。

当該住戸における設計給気風量 $V_{d,SA}$ 及び設計還気風量 $V_{d,RA}$ の代わりに、当該住戸における設計外気風量 $V_{d,OA}$ 及び設計排気風量 $V_{d,EA}$ を用いることができる。

当該住戸における設計給気風量 $V_{d,SA}$ 及び設計還気風量 $V_{d,RA}$ 、又は当該住戸における設計外気風量 $V_{d,OA}$ 及び設計排気風量 $V_{d,EA}$ は、第五章「換気設備」において求めることとする。

A.5 排気過多時における住宅外皮経由の漏気による温度交換効率の補正係数

排気過多時における住宅外皮経由の漏気による温度交換効率の補正係数 C_{leak} は、式(15)により表される値とし、100分の1未満の端数を切り下げた小数第二位までの値とする。ただし、給気と排気の比率による温度交換効率の補正係数 C_{bal} に0.90を用いた場合は、 $C_{leak} = 1.00$ とする。

$$C_{leak} = \begin{cases} \frac{V_{d,SA}}{V_{d,RA}} & (V_{d,RA} > V_{d,SA}) \\ 1 & (V_{d,RA} \leq V_{d,SA}) \end{cases} \quad (15)$$

ここで、

$V_{d,SA}$: 設計給気風量(m³/h)

$V_{d,RA}$: 設計還気風量(m³/h)

である。設計給気風量及び設計還気風量のかわりに設計外気風量及び設計排気風量を用いてもよい。設計給気風量、設計還気風量、設計外気風量及び設計排気風量は、5章「換気設備」に規定される値とする。

(6) 絶対湿度交換効率の把握

上記資料により、風量比がもたらす温度交換能力への影響を評価できた。続いて、絶対湿度交換能力を算出する為に、物質移動速度を説明する。

a. 物質移動速度：ルイス数

ルイス数とは、熱と物質の移動速度の比を表す無次元の物性値である。熱と物質が同時に移動するような系の解析で重要なパラメーターとなる。ルイス数が1の場合、温度と物質移動の分布が相似となる。ルイス数は次式で定義される。

$$Le = \frac{\alpha}{D}$$

式 146

α	: 熱拡散率	$[m^2/s]$
D	: 拡散係数	$[m^2/s]$

また、次式で定義される。

$$Le = \frac{Sc}{Pr}$$

式 147

Pr	: プラントル数
Sc	: シュミット数

以下に温度と湿度を変更した場合のルイス数コンター図を記す。本報告ではルイス数を一定値 0.88 とする。

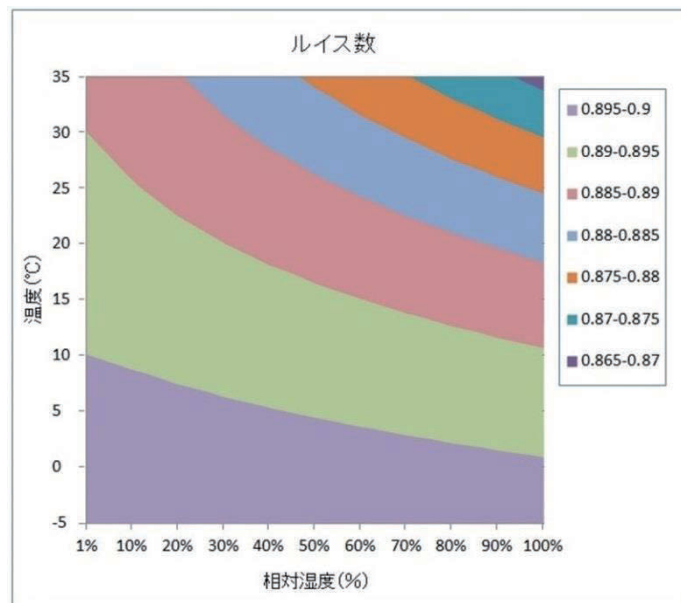


図 239 ルイス数コンター図

3.4.5 全熱交換機温湿度交換効率算出式の作成

全熱交換器の特性式を作成する。計算フローを以下に記す。

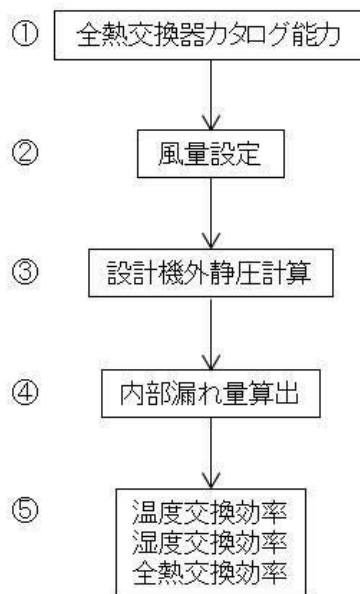


図 240 全熱交換機：交換効率計算フロー

- ① カタログ温度交換効率情報を入手
- ② 給気風量、還気風量の設計
- ③ 据付図面を作成し、OA、SA、RA、EA の機外静圧を算出
- ④ ②～③の風量比、機外静圧より内部漏れ量算出
- ⑤ 内部漏れ量を考慮した温度交換効率、湿度交換効率、全熱交換効率を算出

以下に、詳細計算内容を記す。

<入力項目>

○カタログ効率

温度交換効率： η_c

○風量比より熱交換効率補正係数[-]

α

○ルイス数を用いて湿度交換効率[-]

$$\eta = \eta_c \times \alpha \times Le^3$$

式 148

※ ルイス数を3乗しているのは、エレメントを介して交換を行うので、排気通路⇒エレメント⇒給気通路と3層に分けて物質移動が必要なため。

○温度[°C]

外気： T_{OA}

還気： T_{RA}

給気： T_{SA}' (エレメント通過後)

排気： T_{EA}' (エレメント通過後)

○湿度[g/kg]

外気： X_{OA}

還気： X_{RA}

給気： X_{SA}' (エレメント通過後)

排気： X_{EA}' (エレメント通過後)

○比熱[kJ/kgK]

外気： C_{pOA}

還気： C_{pRA}

給気： C_{pSA}' (エレメント通過後)

排気： C_{pEA}' (エレメント通過後)

○質量流量[kg/s]

給気： Q_{SA}

還気： Q_{RA}

○機外静圧[Pa]

外気： P_{OA}

給気： P_{SA}

還気： P_{RA}

排気： P_{EA}

○ファンモーター発熱[W]

発熱： E_m

<出力項目>

○内部漏れ量算出：RA から SA へ漏れる分[kg/s]

$$Q_{RA \Rightarrow SA}' = (0.001824 \times (P_{RA} - P_{OA})^2 - 0.760524 \times (P_{RA} - P_{OA}) + 68.831334) / 3600$$

式 149

○内部漏れ量算出：OA から EA へ漏れる分[kg/s]

$$Q_{OA \Rightarrow EA}' = (0.001676 \times (P_{RA} - P_{OA})^2 - 0.656959 \times (P_{RA} - P_{OA}) + 60.749704) / 3600$$

式 150

○内部排気側換気量：RA から EA へ流れる分[kg/s]

$$Q_{RA \Rightarrow EA}' = Q_{RA} - Q_{RA \Rightarrow SA}'$$

式 151

○内部給気側換気量：OA から SA へ流れる分[kg/s]

$$Q_{OA \Rightarrow SA}' = Q_{OA} - Q_{OA \Rightarrow EA}'$$

式 152

○内部漏れ量顕熱：RA から SA へ漏れる分[W]

$$E_{RA \Rightarrow SA}' = Q_{RA \Rightarrow SA}' \times C_{pRA} \times T_{RA}$$

式 153

○内部給気側換気量顕熱：OA から SA へ流れる分[W]

$$E_{OA \Rightarrow SA}' = Q_{OA \Rightarrow SA}' \times C_{pSA}' \times T_{SA}'$$

式 154

○RA から SA へ漏れる顕熱と給気側エレメント通過後顕熱合計[W]

$$E_{SA} = E_{RA \Rightarrow SA}' + E_{OA \Rightarrow SA}'$$

式 155

○RA から SA へ漏れる絶対湿度量[g/s]

$$X_{RA \Rightarrow SA}' = Q_{RA \Rightarrow SA}' \times X_{RA}$$

式 156

○OA から SA へ流れる給気側換気量絶対湿度量[g/s]

$$X_{OA \Rightarrow SA}' = Q_{OA \Rightarrow SA}' \times X_{SA}'$$

式 157

○SA 合計絶対湿度量[g/s]

$$X_{SA}' = X_{RA \Rightarrow SA}' + X_{OA \Rightarrow SA}'$$

式 158

○SA 絶対湿度[g/kg]

$$X_{SA} = X_{SA}' / Q_{SA}$$

式 159

○SA 温度[°C]

$$T_{SA} = E_{SA} / Q_{SA} / C_{pSA}$$

式 160

○ファンモーターによる発熱給気温度上昇

$$T_m = E_m / C_{pSA} / Q_{SA}$$

式 161

○ファンモーター入熱を考慮した給気温度

$$T_{SAm} = T_{SA} + T_m$$

式 162

○漏れ込み温度交換効率

$$\eta_t' = (T_{OA} - T_{SAm}) / (T_{OA} - T_{RA})$$

式 163

○内部漏れ量絶対湿度：RA から SA へ漏れる分[g/s]

$$El_{RA \Rightarrow SA} = X_{RA} \times Q_{RA \Rightarrow SA}'$$

式 164

○内部給気側換気量絶対湿度：OA から SA へ流れる分[g/s]

$$El_{OA \Rightarrow SA} = Q_{OA \Rightarrow SA}' \times X_{SA}'$$

式 165

○給気口絶対湿度[g/kg]

$$X_{SA}' = (El_{RA \Rightarrow SA} + El_{OA \Rightarrow SA}) / (Q_{RA \Rightarrow SA}' + Q_{OA \Rightarrow SA}')$$

式 166

○漏れ込み絶対湿度交換効率

$$\eta_x' = (X_{OA} - X_{SA}') / (X_{OA} - X_{RA})$$

式 167

上記計算より、内部漏れ量を考慮した各吸込み口、吹出し口の温湿度が求まる。夏期定格条件と冬期定格条件にて機外静圧を変更した場合温度交換効率、湿度交換効率の変化を記す。

計算条件

○夏期定格条件

外気温度：35℃、外気相対湿度 40%、室内温度：27℃、室内相対湿度 46.6%

○冬期定格条件

外気温度：7℃、外気相対湿度 86.6%、室内温度：20℃、室内相対湿度 58.6%

○機外静圧条件：吸込み口機外静圧差 RA-OA[Pa]

条件 1：90Pa

条件 2：60 Pa

条件 3：30 Pa

条件 4：0Pa(定格条件)

条件 5：-30 Pa

条件 6：-60 Pa

条件 7：-90 Pa

※ 尚、計算条件としてカタログ記載の温度効率を用いて SA と EA 温湿度条件を求めた上で、機外静圧による影響を評価した。また、モーター発熱も考慮している。以下に温湿度交換効率の結果を記す。

表 91 条件別温度交換効率

温度交換 効率[-]	条件 1	条件 2	条件 3	条件 4	条件 5	条件 6	条件 7
夏期	0.552	0.558	0.566	0.575	0.585	0.596	0.609
冬期	0.893	0.898	0.905	0.914	0.924	0.935	0.947

表 92 条件別絶対湿度交換効率

湿度交換 効率[-]	条件 1	条件 2	条件 3	条件 4	条件 5	条件 6	条件 7
夏期	0.531	0.543	0.558	0.576	0.596	0.620	0.646
冬期	0.530	0.542	0.556	0.573	0.593	0.615	0.640

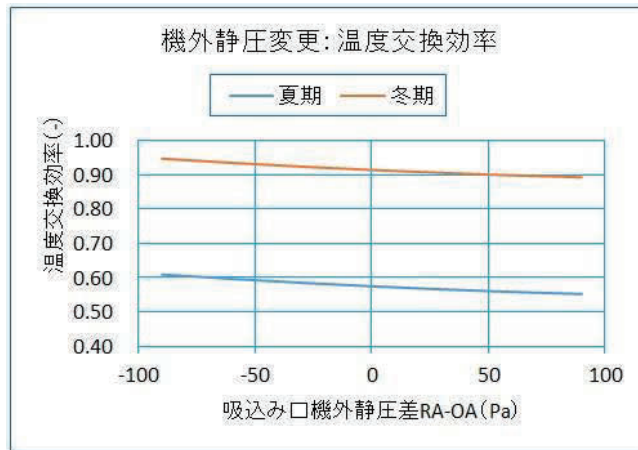


図 241 機外静圧変更時温度交換効率

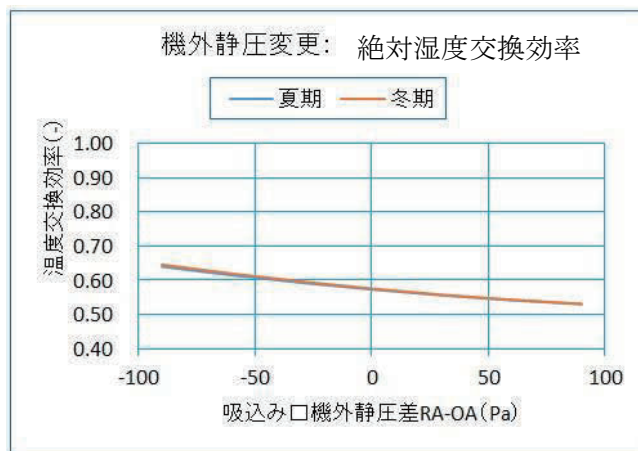


図 242 機外静圧変更時絶対湿度交換効率

上記に温度交換効率と絶対湿度交換効率を記す。吸込み口機外静圧の関係が $RA < OA$ になればなるほど、内部漏れが多くなるので結果的に効率が上がる結果となった。また、冬期に関しては、ファンモーターの入熱が暖房側に働くため、効率が上がる結果となった。尚、絶対湿度交換効率に関しては、ファンモーターの入熱が影響しない為、同一のカーブになったと考えられる。

有効換気量と上記グラフを組み合わせた図を記す。図が示す様に、有効換気量率が低下する（内部漏れ量が増加する）と、RA 空気が給気空気に混ざるの、見かけ上交換効率が増加する結果となった。

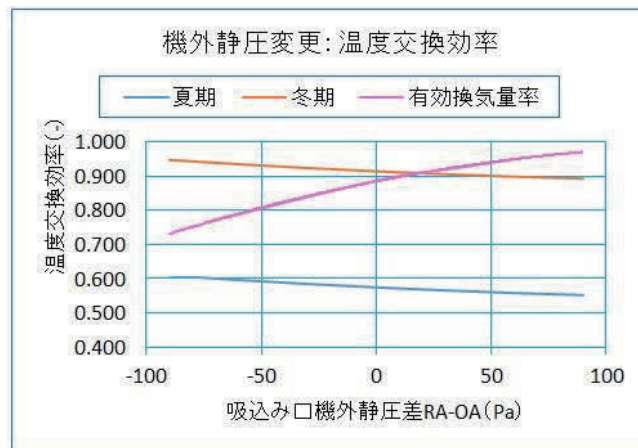


図 243 機外静圧変更時温度交換効率（有効換気量記載）

参考資料：国土交通省：建築設備設計基準を用いて、建築研究所据付に用いた計算結果
 (作成者：ダイキンエアテクノ株式会社)

DESICAシステムの抵抗計算

- ・ DESICA風量：500m³/h、吹出口の風量：500m³/h × 1個
- ・ 選定送風機：HDMP500C (200φ×500m³/h×150Pa×単相200V-1.1kw)

■ EA、RA系統

- A. 局部(深型ヘッドキャブ)圧力損失 [Pa] [風量：500m³/h時]

$$\Delta P = 20.0 \text{ [Pa]}$$
- B. 直み(1/4)外圧力損失 [Pa] ・ダクト長(L)：0.4 [m]、 Δp ：0.5 [Pa/m]
[風量：500m³/h]
[風速：2.8m/s]

$$\Delta P = L \times \Delta p = 0.4 \times 0.5 = 0.2 \text{ [Pa]}$$
- C. 局部(90°エルブ)圧力損失 [Pa] ・損失係数(ζ)：0.21、 ∇ 外内風速(V)：4.42 [m/s]
[風量：500m³/h時]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.21 \times 4.42^2 = 2.5 \text{ [Pa]} \times 1 \text{ 個} = 2.5 \text{ [Pa]}$$
- D. 直み(1/4)外圧力損失 [Pa] ・ダクト長(L)：2.6 [m]、 Δp ：1.5 [Pa/m]
[風量：500m³/h]
[風速：4.4m/s]

$$\Delta P = L \times \Delta p = 2.6 \times 1.5 = 3.9 \text{ [Pa]}$$
- E. 局部(90°エルブ)圧力損失 [Pa] ・損失係数(ζ)：0.22、 ∇ 外内風速(V)：4.42 [m/s]
[風量：500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.22 \times 4.42^2 = 2.6 \text{ [Pa]} \times 2 \text{ 個} = 5.2 \text{ [Pa]}$$
- F. 直み(1/4)外圧力損失 [Pa] ・ダクト長(L)：5.0 [m]、 Δp ：1.5 [Pa/m]
[風量：500m³/h]
[風速：4.4m/s]

$$\Delta P = L \times \Delta p = 5.0 \times 1.5 = 7.5 \text{ [Pa]}$$
- G. 局部(90°エルブ)圧力損失 [Pa] ・損失係数(ζ)：0.22、 ∇ 外内風速(V)：4.42 [m/s]
[風量：500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.22 \times 4.42^2 = 2.6 \text{ [Pa]} \times 2 \text{ 個} = 5.2 \text{ [Pa]}$$
- H. 消音ボックス圧力損失 [Pa] ・損失係数(ζ)：0.83、 ∇ 外内風速(V)：4.42 [m/s]
[風量：500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.83 \times 4.42^2 = 9.7 \text{ [Pa]}$$
- I. 吸込口圧力損失 [Pa] ・損失係数(ζ)：3.00、吹出風速(V)：2.1 [m/s]
[風量：500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 3.0 \times 2.1^2 = 7.94 \text{ [Pa]}$$
- J. 局部(90°エルブ(VD))圧力損失 [Pa] ・損失係数(ζ)：0.20、 ∇ 外内風速(V)：4.42 [m/s]
[風量：500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.2 \times 4.42^2 = 2.3 \text{ [Pa]} \times 1 \text{ 個} = 2.3 \text{ [Pa]}$$
- P (静圧) = 64.4 × 1.20 (余裕率) = 77.3 [Pa]

図 244

■ O A, S A 系統

A. 局部(深型ヘッドキャップ)圧力損失 [Pa] [風量: 500m³/h時]

$$\Delta P = 20.0 \text{ [Pa]}$$

B. 直しのイワシの外圧力損失 [Pa] ・ダクト長(L): 0.4 [m] , Δp: 0.5 [Pa/m]
[風量: 500m³/h]
[風速: 2.8m/s]

$$\Delta P = L \times \Delta p = 0.4 \times 0.5 = 0.2 \text{ [Pa]}$$

C. 局部(おん)の圧力損失 [Pa] ・損失係数(ζ): 0.20, 9°外内風速(V): 4.42 [m/s]
[風量: 500m³/h時]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.2 \times 4.42^2 = 2.3 \text{ [Pa]} \times 1 \text{ 個} = 2.3 \text{ [Pa]}$$

D. 直しのイワシの外圧力損失 [Pa] ・ダクト長(L): 4.3 [m] , Δp: 1.5 [Pa/m]
[風量: 500m³/h]
[風速: 4.4m/s]

$$\Delta P = L \times \Delta p = 4.3 \times 1.5 = 6.5 \text{ [Pa]}$$

E. 局部(90°I&I)の圧力損失 [Pa] ・損失係数(ζ): 0.22, 9°外内風速(V): 4.42 [m/s]
[風量: 500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.22 \times 4.42^2 = 2.6 \text{ [Pa]} \times 4 \text{ 個} = 10.3 \text{ [Pa]}$$

F. 直しのイワシの外圧力損失 [Pa] ・ダクト長(L): 1.5 [m] , Δp: 1.5 [Pa/m]
[風量: 500m³/h]
[風速: 4.4m/s]

$$\Delta P = L \times \Delta p = 1.5 \times 1.5 = 2.3 \text{ [Pa]}$$

G. 局部(90°I&I)の圧力損失 [Pa] ・損失係数(ζ): 0.22, 9°外内風速(V): 4.42 [m/s]
[風量: 500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.22 \times 4.42^2 = 2.6 \text{ [Pa]} \times 2 \text{ 個} = 5.2 \text{ [Pa]}$$

H. 消音ボックスの圧力損失 [Pa] ・損失係数(ζ): 0.83, 9°外内風速(V): 4.42 [m/s]
[風量: 500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.83 \times 4.42^2 = 9.7 \text{ [Pa]}$$

I. 吹出口の圧力損失 [Pa] ・損失係数(ζ): 3.60, 吹出風速(V): 2.1 [m/s]
[風量: 500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 3.6 \times 2.1^2 = 9.53 \text{ [Pa]}$$

J. 局部(おん)-(VD)の圧力損失 [Pa] ・損失係数(ζ): 0.20, 9°外内風速(V): 4.42 [m/s]
[風量: 500m³/h]

$$\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.2 \times 4.42^2 = 2.3 \text{ [Pa]} \times 1 \text{ 個} = 2.3 \text{ [Pa]}$$

$$P(\text{静圧}) = 68.3 \times 1.20 (\text{余裕率}) = 82.0 \text{ [Pa]}$$

図 245

◎VKMP (外気処理機) 系統の抵抗計算

- ・VKMP風量：500m³/h，吹出口の風量：500m³/h × 1個
- ・選定送風機：VKP50HM (200φ*500m³/h*200Pa*单相200V-0.56kw)

■ E.A. RA系統

- A. 局部(深型ヘッドキャブ)圧力損失 [Pa] [風量：500m³/h時]
 $\Delta P = 20.0$ [Pa]
- B. 直入(15M)外圧力損失 [Pa] ・ダクト長(L)：0.4 [m]， Δp ：0.5 [Pa/m]
 $\Delta P = L \times \Delta p = 0.4 \times 0.5 = 0.2$ [Pa] [風量：500m³/h]
[風速：2.8m/s]
- C. 局部(15M)圧力損失 [Pa] ・損失係数(ζ)：0.21， ∇ 外内風速(V)：4.42 [m/s]
 $\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.21 \times 4.42^2 = 2.5$ [Pa] × 1個 = 2.5 [Pa] [風量：500m³/h時]
- D. 直入(15M)外圧力損失 [Pa] ・ダクト長(L)：6.1 [m]， Δp ：1.5 [Pa/m]
 $\Delta P = L \times \Delta p = 6.1 \times 1.5 = 9.2$ [Pa] [風量：500m³/h]
[風速：4.4m/s]
- E. 局部(90°15M)圧力損失 [Pa] ・損失係数(ζ)：0.22， ∇ 外内風速(V)：4.42 [m/s]
 $\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.22 \times 4.42^2 = 2.6$ [Pa] × 3個 = 7.7 [Pa] [風量：500m³/h]
- F. 直入(15M)外圧力損失 [Pa] ・ダクト長(L)：6.4 [m]， Δp ：1.5 [Pa/m]
 $\Delta P = L \times \Delta p = 6.4 \times 1.5 = 9.6$ [Pa] [風量：500m³/h]
[風速：4.4m/s]
- G. 局部(90°15M)圧力損失 [Pa] ・損失係数(ζ)：0.22， ∇ 外内風速(V)：4.42 [m/s]
 $\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.22 \times 4.42^2 = 2.6$ [Pa] × 4個 = 10.3 [Pa] [風量：500m³/h]
- H. 消音ボックス圧力損失 [Pa] ・損失係数(ζ)：0.83， ∇ 外内風速(V)：4.42 [m/s]
 $\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.83 \times 4.42^2 = 9.7$ [Pa] [風量：500m³/h]
- I. 吸込口圧力損失 [Pa] ・損失係数(ζ)：3.00，吹出風速(V)：2.1 [m/s]
 $\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 3.0 \times 2.1^2 = 7.94$ [Pa] [風量：500m³/h]
- J. 局部(15M)-(VD)圧力損失 [Pa] ・損失係数(ζ)：0.20， ∇ 外内風速(V)：4.42 [m/s]
 $\Delta P = 0.6 \times \zeta \times V^2 = 0.6 \times 0.2 \times 4.42^2 = 2.3$ [Pa] × 1個 = 2.3 [Pa] [風量：500m³/h]
- P (静圧) = 79.5 × 1.20 (余裕率) = 95.4 [Pa]

図 246

■ O.A. SA系統

- A. 局部(深型ヘッドキャップ)圧力損失 [Pa] [風量: 500m³/h時]
 $\Delta P = 20.0$ [Pa]
- B. 直しのイワリ外圧力損失 [Pa] ・ダクト長(L): 0.4 [m] , Δp : 0.5 [Pa/m]
 [風量: 500m³/h]
 [風速: 2.8m/s]
 $\Delta P = L \times \Delta p$ [Pa]
 $= 0.4 \times 0.5 = 0.2$ [Pa]
- C. 局部(ホソ)圧力損失 [Pa] ・損失係数(ζ): 0.20 , 外内風速(V): 4.42 [m/s]
 [風量: 500m³/h時]
 $\Delta P = 0.6 \times \zeta \times V^2$ [Pa]
 $= 0.6 \times 0.2 \times 4.42^2 = 2.3$ [Pa] × 1 個 = 2.3 [Pa]
- D. 直しのイワリ外圧力損失 [Pa] ・ダクト長(L): 5.3 [m] , Δp : 1.5 [Pa/m]
 [風量: 500m³/h]
 [風速: 4.4m/s]
 $\Delta P = L \times \Delta p$ [Pa]
 $= 5.3 \times 1.5 = 8.0$ [Pa]
- E. 局部(90° 曲)圧力損失 [Pa] ・損失係数(ζ): 0.22 , 外内風速(V): 4.42 [m/s]
 [風量: 500m³/h]
 $\Delta P = 0.6 \times \zeta \times V^2$ [Pa]
 $= 0.6 \times 0.22 \times 4.42^2 = 2.6$ [Pa] × 1 個 = 2.6 [Pa]
- F. 直しのイワリ外圧力損失 [Pa] ・ダクト長(L): 2.0 [m] , Δp : 1.5 [Pa/m]
 [風量: 500m³/h]
 [風速: 4.4m/s]
 $\Delta P = L \times \Delta p$ [Pa]
 $= 2.0 \times 1.5 = 3.0$ [Pa]
- G. 局部(90° 曲)圧力損失 [Pa] ・損失係数(ζ): 0.22 , 外内風速(V): 4.42 [m/s]
 [風量: 500m³/h]
 $\Delta P = 0.6 \times \zeta \times V^2$ [Pa]
 $= 0.6 \times 0.22 \times 4.42^2 = 2.6$ [Pa] × 2 個 = 5.2 [Pa]
- H. 消音ボックス圧力損失 [Pa] ・損失係数(ζ): 0.83 , 外内風速(V): 4.42 [m/s]
 [風量: 500m³/h]
 $\Delta P = 0.6 \times \zeta \times V^2$ [Pa]
 $= 0.6 \times 0.83 \times 4.42^2 = 9.7$ [Pa]
- I. 吹出口圧力損失 [Pa] ・損失係数(ζ): 3.60 , 吹出風速(V): 2.1 [m/s]
 [風量: 500m³/h]
 $\Delta P = 0.6 \times \zeta \times V^2$ [Pa]
 $= 0.6 \times 3.6 \times 2.1^2 = 9.53$ [Pa]
- J. 局部(ホソ)圧力損失 [Pa] ・損失係数(ζ): 0.20 , 外内風速(V): 4.42 [m/s]
 [風量: 500m³/h]
 $\Delta P = 0.6 \times \zeta \times V^2$ [Pa]
 $= 0.6 \times 0.2 \times 4.42^2 = 2.3$ [Pa] × 1 個 = 2.3 [Pa]
- P (静圧) = 62.8 × 1.20 (余裕率) = 75.4 [Pa]

図 247

3.5 ヒートポンプデシカント調湿外調機特性の把握

3.5.1 適用範囲

居住空間の空気調和における換気と調湿の機能を併せ持ち、吸着材料の吸着、再生にヒートポンプの熱源を用いるとともに、空気通路および冷媒回路のバッチ切換えによって連続除加湿運転を可能とするヒートポンプデシカント調湿外調機について記載する。

3.5.2 カタログ（技術資料）表記項目

ヒートポンプデシカント調湿外調機およびヒートポンプデシカント調湿外調機と組み合わせて潜熱・顕熱分離空調を実現する高顕熱形の空調機の機種ラインアップを表 93 に示す。

表 93 ラインアップ

機器名称		ラインアップ
ヒートポンプデシカント調湿外調機		250,500,1500,2000m ³ /h
高顕熱形	熱源機	8～54HP (2HP 毎)
ビル用マルチ	室内ユニット	2.2,2.8,3.6,4.5,5.6,7.1,8.0,9.0,11.2,14.0,16.0,22.4,28.0kW

3.5.3 標準仕様

ヒートポンプデシカント調湿外調機のカatalog記載の性能、性能評価条件、引用規格を表 94 に示す。

表 94 ヒートポンプデシカント調湿外調機カatalog表記値、引用規格

			評価条件	評価	引用規格	
機種			—	—	—	
構成機種			—	—	JISB8628	
電源		相,電圧 周波数	—	—	JISB8616	
加湿量		kg/h	室内条件：22°C D.B, 50% RH 室外条件：0°C D.B, 50% RH 定格能力： 外気潜熱負荷を処理する際の能力 全熱,顕熱能力は最大能力も併せて併記	97% 以上	ビル衛生管理 講習会資料 (p11) (H20 年度) 東京都健康安全研究 センター (4)空気線図を使った 加湿計算の実例より (室内,室外条件)	
除湿 冷房 性能	全熱能力	kW	室内条件：27°C D.B, 19°C W.B 室外条件：33°C D.B, 28°C W.B 定格能力： 外気潜熱負荷を処理する際の能力 全熱,顕熱能力は最大能力も併せて併記	97% 以上	JISB8616	
	顕熱能力	W		105% 以下		
	消費電力	W				
加湿 暖房 性能	全熱能力	kW	室内条件：22°C D.B, 50% RH 室外条件：0°C D.B, 50% RH 定格能力： 外気潜熱負荷を処理する際の能力 全熱,顕熱能力は最大能力も併せて併記	97% 以上	JISB8616 ビル衛生管理 講習会資料 (p11) (H20 年度) 東京都健康安全研究 センター (4)空気線図を使った 加湿計算の実例より	
	顕熱能力	W		105% 以下		
	消費電力	W				
空調 定格点 冷房性能	全熱能力	kW	室内条件：27°C D.B, 19°C W.B 室外条件：35°C D.B, 24°C W.B 定格能力：外気潜熱負荷を処理する際の能力 全熱,顕熱能力は最大能力も併せて併記	97% 以上	JISB8616	
	顕熱能力	W		105% 以下		
	消費電力	W				
空調 定格点 暖房性能	全熱能力	kW	室内条件：20°C D.B, 13.8°C W.B 室外条件：7°C D.B, 6°C W.B 定格能力： 外気潜熱負荷を処理する際の能力 全熱,顕熱能力は最大能力も併せて併記	97% 以上	JISB8616	
	顕熱能力	W		105% 以下		
	消費電力	W				
外装		材質	(亜鉛鋼板)	—	—	
外形寸法	高さ×幅×奥行	mm	—	—	—	
熱交換器			—	—	—	
圧縮機	型式		—	—	—	
	電動機出力×台数	kW	—	—	—	
	始動方式		—	—	—	
ファン	型式		—	—	—	
	電動機出力×台数	kW	—	—	—	
	定格風量	m ³ /h	—	±10% 以内	JISB8616 JISB8628	
	駆動方式		—	—	—	
	定格機外静圧	Pa	—	±10% 以内	JISB8616 JISB8628	
質量		kg	—	—	JISB8616	
運転音 (A スケール)		dB	—	+2db 以内	JISB8616	
容量制御		%	—	—	—	
冷媒 (R410A) 充填量		kg	—	—	JISB8616	
冷媒制御		電子膨張弁			—	
法定冷凍トン		—			—	冷凍保安規則

3.5.4 ヒートポンプデシカント調湿外調機性能特性

ヒートポンプデシカント調湿外調機のカatalogに記載されている性能特性を表 95, 表 96 に示す。
表に示すように、室外温湿度、風量変化時の最大全熱能力、最大顕熱能力特性、消費電力が記載されている。

これらの機種選定に必要な性能特性が明らかにされている一方、年間消費電力試算に必要な部分負荷時の性能については記載されていない。

表 95 除湿冷房最大能力特性

外気温度 °CDB 相対湿度 %RH	除湿冷房	機種	能力種別	kW	室内吸込状態温度 °CDB 相対湿度 %RH																				
					23.0			24.0			25.0			26.0			27.0			28.0					
					45%RH	55%RH	65%RH	45%RH	55%RH	65%RH	45%RH	55%RH	65%RH	45%RH	55%RH	65%RH	45%RH	55%RH	65%RH	45%RH	55%RH	65%RH			
20.0	60W	全熱能力	顕熱能力	5.18	4.09	4.09	4.19	4.09	3.90	4.01	3.91	3.89	3.97	3.81	3.79	3.84	3.72	3.69	3.75	3.62	3.68	3.65	3.52	3.58	
			消費電力 (全熱能力/kW)	0.35	0.35	0.35	0.31	0.31	0.32	0.32	0.32	0.33	0.33	0.34	0.34	0.35	0.35	0.36	0.36	0.36	0.37	0.37	0.37	0.37	0.37
	60W	全熱能力	顕熱能力	4.32	4.42	4.33	4.43	4.34	4.24	4.34	4.24	4.14	4.24	4.15	4.04	4.14	4.04	3.93	4.04	3.92	3.89	3.81	3.78	3.67	
			消費電力 (全熱能力/kW)	0.89	0.89	0.89	0.84	0.84	0.84	0.79	0.79	0.79	0.74	0.74	0.74	0.69	0.69	0.69	0.64	0.64	0.64	0.59	0.59	0.59	0.69
	25.0	60W	全熱能力	顕熱能力	1.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
				消費電力 (全熱能力/kW)	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	27.0	60W	全熱能力	顕熱能力	6.79	6.79	6.79	6.74	6.74	6.74	6.69	6.69	6.69	6.64	6.64	6.64	6.59	6.59	6.59	6.54	6.54	6.54	6.49	6.49	6.29
				消費電力 (全熱能力/kW)	1.24	1.24	1.25	1.25	1.25	1.25	1.27	1.27	1.28	1.28	1.28	1.29	1.29	1.29	1.30	1.31	1.32	1.32	1.32	1.32	1.32
		60W	全熱能力	顕熱能力	4.84	4.75	4.65	4.78	4.68	4.58	4.67	4.57	4.46	4.56	4.47	4.35	4.45	4.35	4.23	4.33	4.23	4.15	4.11	4.04	3.84
				消費電力 (全熱能力/kW)	0.89	0.89	0.89	0.83	0.83	0.83	0.78	0.78	0.78	0.73	0.73	0.73	0.68	0.68	0.68	0.63	0.63	0.63	0.58	0.58	0.58
		40W	全熱能力	顕熱能力	1.27	1.27	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
				消費電力 (全熱能力/kW)	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
29.0		60W	全熱能力	顕熱能力	6.87	6.87	6.87	6.82	6.82	6.82	6.77	6.77	6.77	6.72	6.72	6.72	6.67	6.67	6.67	6.62	6.62	6.62	6.57	6.57	6.37
				消費電力 (全熱能力/kW)	1.21	1.22	1.22	1.23	1.23	1.24	1.24	1.24	1.25	1.25	1.26	1.26	1.27	1.27	1.28	1.28	1.29	1.29	1.29	1.29	1.29
		60W	全熱能力	顕熱能力	4.74	4.18	4.09	4.13	4.03	3.99	4.02	3.94	3.86	3.92	3.83	3.74	3.81	3.72	3.62	3.79	3.69	3.59	3.49	3.48	3.28
				消費電力 (全熱能力/kW)	1.19	1.29	1.29	1.15	1.15	1.15	1.10	1.10	1.05	1.05	1.05	1.00	1.00	1.00	0.95	0.95	0.95	0.90	0.90	0.90	0.90
		40W	全熱能力	顕熱能力	5.18	5.09	5.09	5.10	5.09	4.99	5.02	4.91	4.89	4.82	4.79	4.64	4.72	4.64	4.53	4.69	4.58	4.48	4.38	4.38	4.18
				消費電力 (全熱能力/kW)	1.07	1.07	1.07	1.02	1.02	1.02	0.97	0.97	0.93	0.93	0.93	0.88	0.88	0.88	0.83	0.83	0.83	0.78	0.78	0.78	0.78
	31.0	60W	全熱能力	顕熱能力	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	
				消費電力 (全熱能力/kW)	0.86	0.79	0.68	0.78	0.68	0.58	0.69	0.59	0.48	0.61	0.59	0.52	0.49	0.58	0.49	0.39	0.48	0.39	0.29	0.18	0.34
		60W	全熱能力	顕熱能力	0.84	0.95	0.95	0.93	0.93	0.93	0.85	0.85	0.85	0.80	0.80	0.80	0.75	0.75	0.75	0.70	0.70	0.65	0.65	0.65	0.65
				消費電力 (全熱能力/kW)	1.19	1.19	1.19	1.20	1.20	1.21	1.21	1.22	1.22	1.22	1.23	1.23	1.24	1.24	1.25	1.25	1.26	1.26	1.27	1.28	1.28
		40W	全熱能力	顕熱能力	4.59	4.51	4.43	4.44	4.40	4.31	4.27	4.23	4.19	4.25	4.16	4.06	4.14	4.04	3.94	4.03	3.92	3.81	3.71	3.79	3.69
				消費電力 (全熱能力/kW)	1.29	1.29	1.29	1.29	1.31	1.31	1.32	1.32	1.33	1.33	1.34	1.34	1.35	1.35	1.36	1.37	1.37	1.38	1.38	1.38	1.38
33.0		60W	全熱能力	顕熱能力	5.55	5.46	5.37	5.47	5.37	5.27	5.38	5.28	5.17	5.29	5.19	5.07	5.21	5.09	4.97	5.12	4.99	4.87	5.03	4.89	4.78
				消費電力 (全熱能力/kW)	1.18	1.18	1.18	1.11	1.11	1.11	1.06	1.06	1.01	1.01	1.01	0.96	0.96	0.96	0.91	0.92	0.92	0.87	0.87	0.87	0.87
		60W	全熱能力	顕熱能力	1.02	1.02	1.02	0.97	0.97	0.97	0.92	0.92	0.92	0.87	0.87	0.87	0.82	0.82	0.82	0.77	0.77	0.77	0.72	0.72	
				消費電力 (全熱能力/kW)	1.15	1.14	1.14	1.17	1.17	1.18	1.18	1.19	1.19	1.19	1.20	1.20	1.21	1.21	1.22	1.22	1.23	1.23	1.23	1.24	1.24
		40W	全熱能力	顕熱能力	4.96	4.87	4.78	4.84	4.75	4.69	4.73	4.62	4.52	4.43	4.39	4.29	4.49	4.37	4.25	4.37	4.24	4.12	4.23	4.09	3.96
				消費電力 (全熱能力/kW)	1.49	1.49	1.49	1.35	1.35	1.35	1.30	1.30	1.25	1.25	1.25	1.20	1.20	1.20	1.15	1.15	1.15	1.10	1.10	1.10	1.10
	35.0	60W	全熱能力	顕熱能力	1.27	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
				消費電力 (全熱能力/kW)	0.99	0.89	0.79	0.87	0.77	0.67	0.88	0.67	0.57	0.69	0.58	0.47	0.69	0.49	0.39	0.51	0.39	0.29	0.25	0.42	0.15
		60W	全熱能力	顕熱能力	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	1.74	
				消費電力 (全熱能力/kW)	1.29	1.29	1.29	1.21	1.21	1.22	1.22	1.23	1.23	1.24	1.24	1.25	1.25	1.26	1.26	1.26	1.27	1.27	1.27	1.28	1.28
		40W	全熱能力	顕熱能力	6.81	6.72	6.63	6.73	6.63	6.53	6.64	6.54	6.43	6.55	6.45	6.33	6.47	6.35	6.23	6.38	6.25	6.13	6.29	6.15	6.02
				消費電力 (全熱能力/kW)	1.08	1.08	1.08	1.03	1.03	1.03	0.98	0.98	0.93	0.93	0.93	0.88	0.88	0.88	0.83	0.83	0.83	0.78	0.78	0.78	0.78
37.0		60W	全熱能力	顕熱能力	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	
				消費電力 (全熱能力/kW)	1.08	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09
		60W	全熱能力	顕熱能力	5.72	5.69	5.49	5.59	5.47	5.34	5.46	5.33	5.23	5.19	5.05	5.18	5.03	4.89	5.09	4.85	4.69	4.83	4.65	4.50	
				消費電力 (全熱能力/kW)	1.39	1.39	1.39	1.34	1.34	1.34	1.29	1.29	1.24	1.24	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19
		40W	全熱能力	顕熱能力	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	
				消費電力 (全熱能力/kW)	1.39	1.39	1.39	1.34	1.34	1.34	1.29	1.29	1.24	1.24	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19
	60W	全熱能力	顕熱能力	1.19	1.19	1.19	1.14	1.14	1.14	1.09	1.09	1.09	1.04	1.04	1.04	0.99	0.99	0.94	0.94	0.89	0.89	0.89	0.89		
			消費電力 (全熱能力/kW)	1.04	1.04	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	

外気温度 °CDB 相対湿度 %RH	都市名	温度	湿度	能力及び消費電力	室内吸込状態温度 °CDB 相対湿度 %RH					
					26°C		27°C		28°C	
					45%	55%	45%	55%	45%	55%
20.0	東京	34.4	56.4	全熱能力	5.73	5.61	5.64	5.52	5.55	5.42
				顕熱能力	1.13	1.13	1.08	1.08	1.03	1.03
				消費電力	1.24	1.25	1.26	1.26	1.27	1.28
				全熱能力	5.89	5.77	5.80	5.67	5.66	5.49
				顕熱能力	1.23	1.24	1.19	1.19	1.13	1.13
				消費電力	1.24	1.24	1.25	1.25	1.25	1.25
	名古屋	36.1	52.4	全熱能力	5.81	5.69	5.72	5.60	5.63	5.50
				顕熱能力	1.17	1.17	1.12	1.12	1.07	1.07
				消費電力	1.24	1.24	1.25	1.25	1.27	1.27
				全熱能力	5.84	5.72	5.75	5.62	5.66	5.52
				顕熱能力	1.13	1.13	1.08	1.08	1.03	1.03
				消費電力	1.23	1.24	1.25	1.25	1.26	1.27
27.0	大阪	35.1	55.0	全熱能力	5.87	5.75	5.78	5.65	5.68	5.55
				顕熱能力	1.11	1.12	1.06	1.06	1.01	1.01
				消費電力	1.23	1.24	1.25	1.25	1.26	1.27
				全熱能力	5.84	5.72	5.75	5.62	5.66	5.52
				顕熱能力	1.13	1.13	1.08	1.08	1.03	1.03
				消費電力	1.23	1.24	1.25	1.25	1.26	1.27
	広島	34.6	57.8	全熱能力	5.87	5.75	5.78	5.65	5.68	5.55
				顕熱能力	1.11	1.12	1.06	1.06	1.01	1.01
				消費電力	1.23	1.24	1.25	1.25	1.26	

3.5.5 ヒートポンプデシカント調湿外調機性能計算

前項に示すように、性能計算を実施する上で必要となる負荷率が変化した際の機器特性はカタログおよび技術資料に記載されていない。そこで、以下に示す性能特性式の作成を行った。

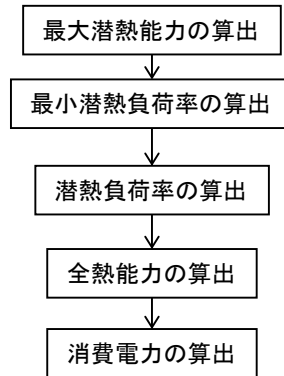


図 248 ヒートポンプデシカント調湿外調機計算フロー

3.5.5.1 最大潜熱能力

室内、室外空気条件における最大潜熱能力を下式で示す。

$\alpha_{00} \sim \alpha_{14}$: 最大潜熱能力係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
LC_{max}	: 最大潜熱能力	[W]

$$\text{最大潜熱能力 (LC}_{max}\text{)} = \alpha_{00} + \alpha_{11} \times T_{OA} + \alpha_{12} \times X_{OA} + \alpha_{13} \times T_{RA} + \alpha_{14} \times X_{RA}$$

式 168

3.5.5.2 最小潜熱負荷率

最小潜熱負荷率を下式で表す。

$\beta_{00} \sim \beta_{24}$: 最小潜熱負荷率係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
$PLRL_{min}$: 最小潜熱負荷率	

$$\begin{aligned} \text{最小潜熱負荷率} = & \beta_{00} + \beta_{11} \times T_{OA} + \beta_{12} \times X_{OA} + \beta_{13} \times T_{RA} + \beta_{14} \times X_{RA} + \beta_{21} \times T_{OA}^2 \\ & + \beta_{22} \times X_{OA}^2 + \beta_{23} \times T_{RA}^2 + \beta_{24} \times X_{RA}^2 \end{aligned}$$

式 169

3.5.5.3 潜熱負荷率

潜熱負荷率を下式で定義する。

LC_{max}	: 最大潜熱能力	[W]
LC	: 潜熱能力 (負荷)	[W]
PLRL	: 潜熱負荷率	

$$\text{潜熱負荷率 (PLRL)} = \text{潜熱負荷 (LC)} / \text{最大潜熱能力 (LC}_{max}\text{)}$$

式 170

3.5.5.4 全熱能力

全熱能力を下式で表す。

$\gamma_{00} \sim \gamma_{34}$: 全熱能力係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
PLRL	: 潜熱負荷率 < 最小潜熱負荷率以上、1 以下の範囲に制限される >	

$$\begin{aligned} \text{全熱能力} = & \gamma_{00} + \gamma_{11} \times T_{OA} + \gamma_{12} \times X_{OA} + \gamma_{13} \times T_{RA} + \gamma_{14} \times X_{RA} + \gamma_{15} \times \text{PLRL} + \gamma_{21} \times T_{OA} \times \\ & \text{PLRL} + \alpha_{22} \times X_{OA} \times \text{PLRL} + \alpha_{23} \times T_{RA} \times \text{PLRL} + \alpha_{24} \times X_{RA} \times \text{PLRL} + \alpha_{31} \times T_{OA} \times \\ & \text{PLRL}^2 + \alpha_{32} \times X_{OA} \times \text{PLRL}^2 + \alpha_{33} \times T_{RA} \times \text{PLRL}^2 + \alpha_{34} \times X_{RA} \times \text{PLRL}^2 \end{aligned}$$

式 171

3.5.5.5 消費電力

消費電力を下式で表す。

$\delta_{00} \sim \delta_{34}$: 消費電力係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
PLRL	: 潜熱負荷率 < 最小潜熱負荷率以上、1 以下の範囲に制限される >	

$$\begin{aligned} \text{消費電力} = & \delta_{00} + \delta_{11} \times T_{OA} + \delta_{12} \times X_{OA} + \delta_{13} \times T_{RA} + \delta_{14} \times X_{RA} + \delta_{15} \times \text{PLRL} + \delta_{21} \times T_{OA} \times \\ & \text{PLRL} + \delta_{22} \times X_{OA} \times \text{PLRL} + \delta_{23} \times T_{RA} \times \text{PLRL} + \delta_{24} \times X_{RA} \times \text{PLRL} + \delta_{31} \times T_{OA} \times \\ & \text{PLRL}^2 + \delta_{32} \times X_{OA} \times \text{PLRL}^2 + \delta_{33} \times T_{RA} \times \text{PLRL}^2 + \delta_{34} \times X_{RA} \times \text{PLRL}^2 \end{aligned}$$

式 172

3.5.6 実測データとの比較検証

3.5.6.1 東京 A ビル（夏季）

東京 A ビルに設置されているヒートポンプデシカント調湿外調機の 6/2～9/11 の期間における、運転データの分析および性能特性式から算出される全熱能力，消費電力と実測値の比較を行った。図 249 に能力の実測値と性能特性式から算出した値の比較、図 250 に消費電力量の実測値と性能特性式から算出した値の比較を示す。

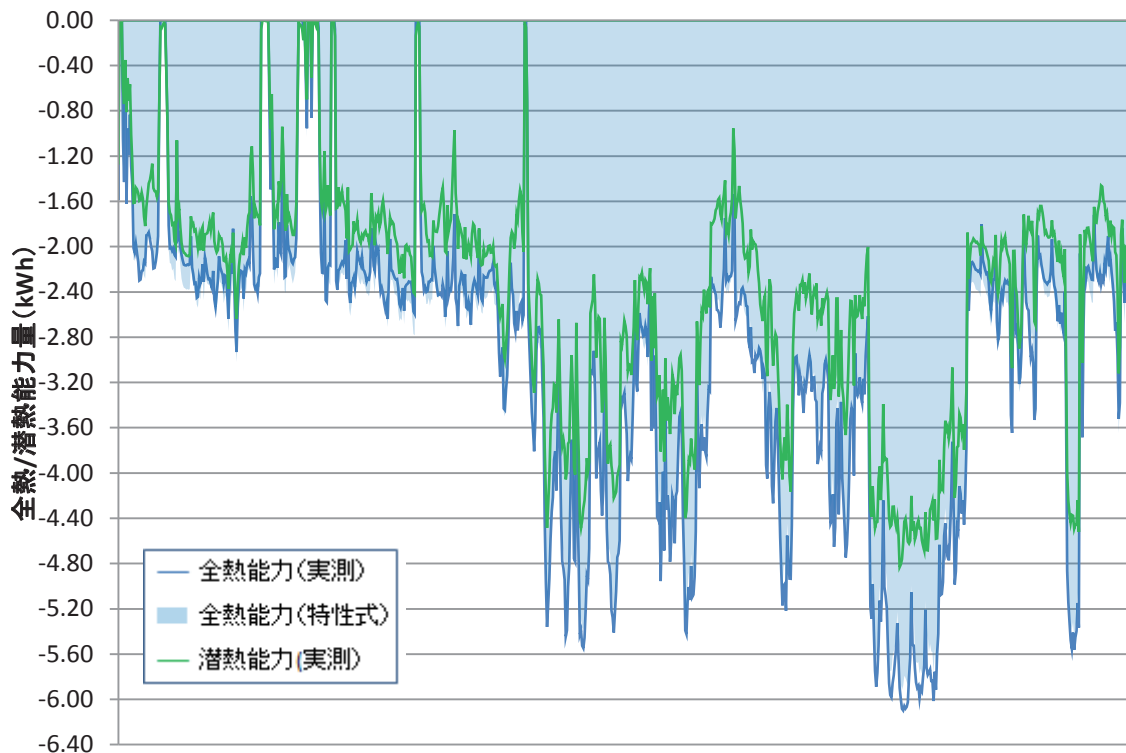


図 249 夏季期間中の特性式を用いた計算値と実測値の比較（能力）

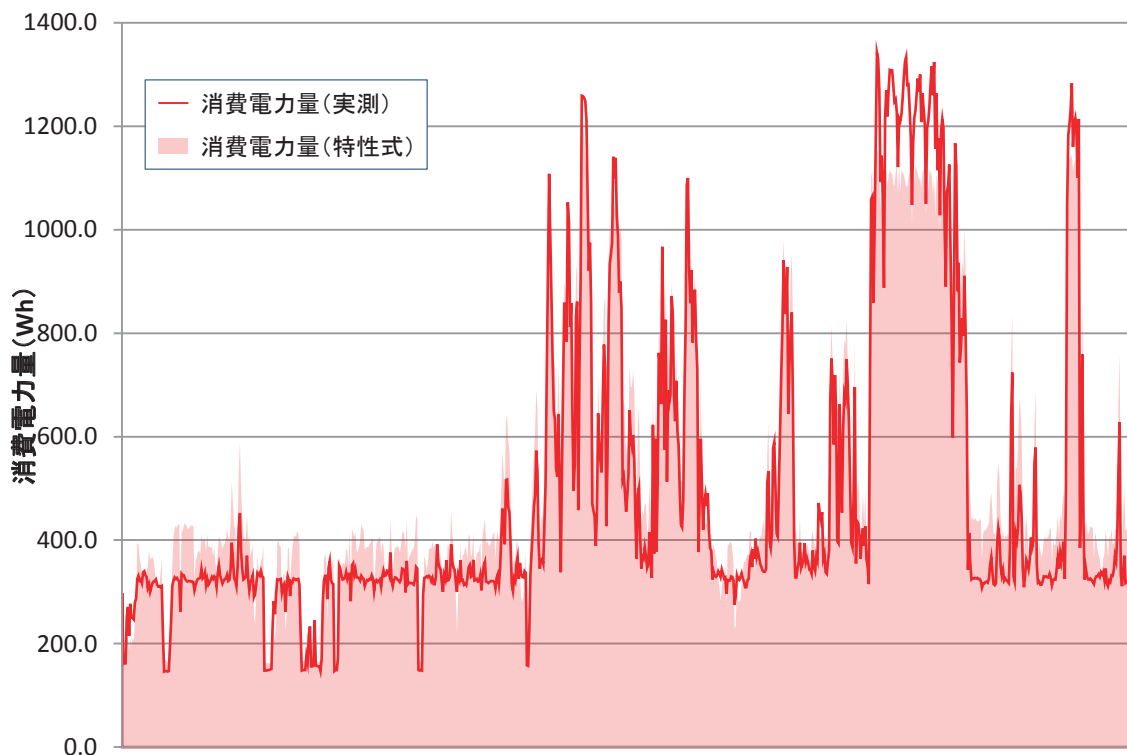


図 250 夏季期間中の特性式を用いた計算値と実測値の比較（消費電力量）

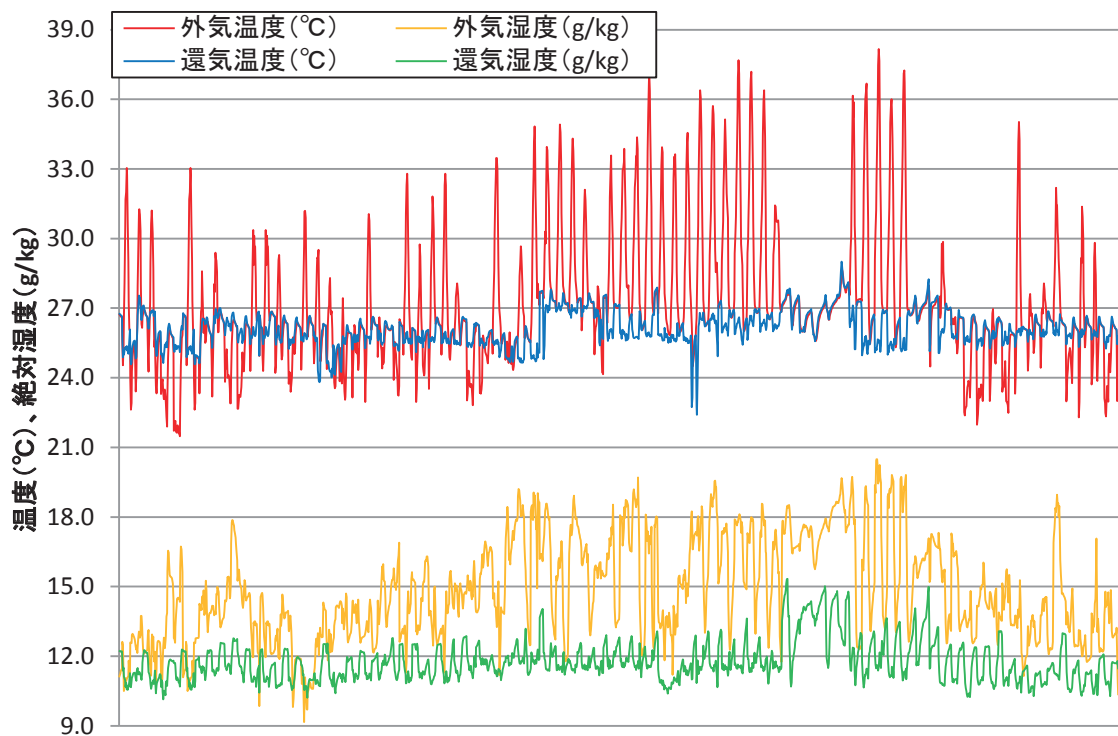


図 251 夏期期間中の外気温湿度、室内温湿度

表 97 夏季分析結果

	実測	計算	誤差
全熱能力期間積算値[kWh]	-2574.5	-2602.9	1.10%
潜熱能力期間積算値[kWh]	-2063.6	—	—
消費電力期間積算値[kWh]	424.8	459.2	8.09%
COP	6.06	—	—
COP(潜熱)	4.86	—	—

期間積算で全熱能力の誤差が 1%強、消費電力量の誤差が 8%強となった。

消費電力の誤差は、負荷が比較的小さい条件において性能特性式の方が実測値と比較して高い消費電力値を示すことが要因であると考えられる。これは、性能特性を作成する際に低い負荷率の取得データが不十分であったことが原因と考えられる。

3.5.6.2 東京 A ビル（冬季）

冬期においては、1/30～4/18 期間にて分析を行った。

図 252 に能力の実測値と性能特性式から算出した値の比較、図 253 に消費電力量の実測値と性能特性式から算出した値の比較を示す。

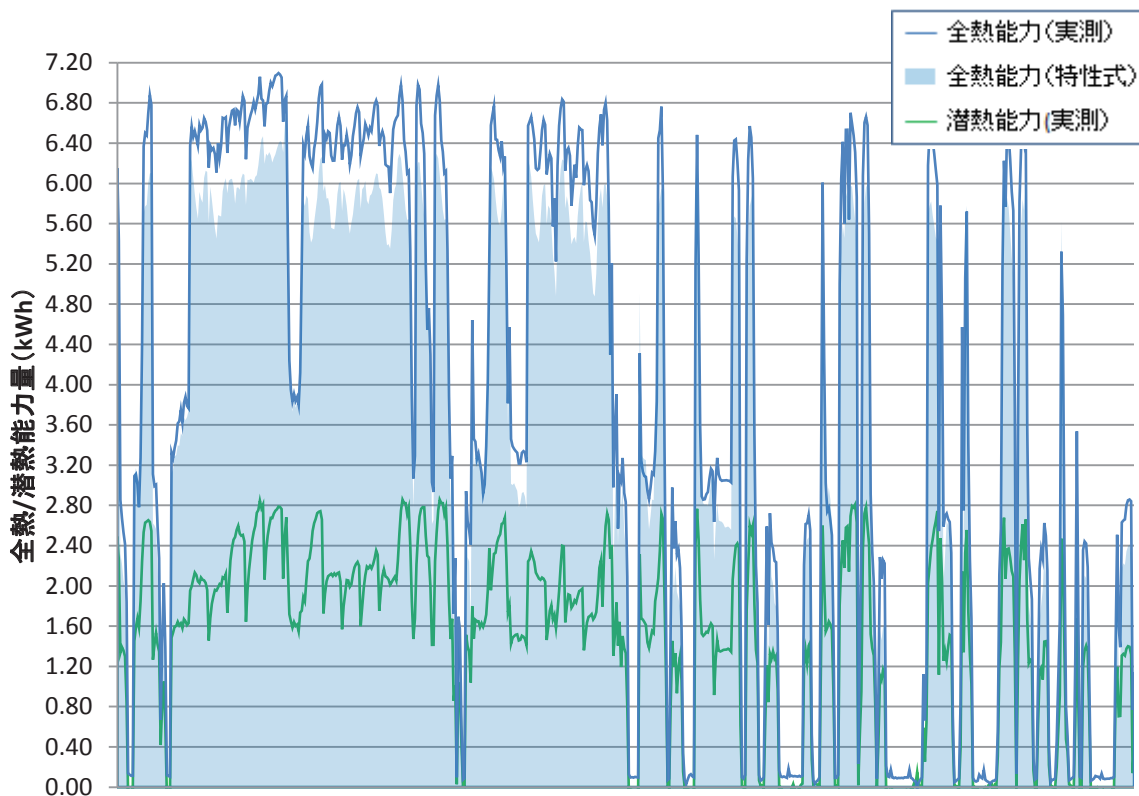


図 252 冬季期間中の特性式と実測値の比較（能力）



図 253 冬季期間中の特性式と実測値の比較（消費電力量）

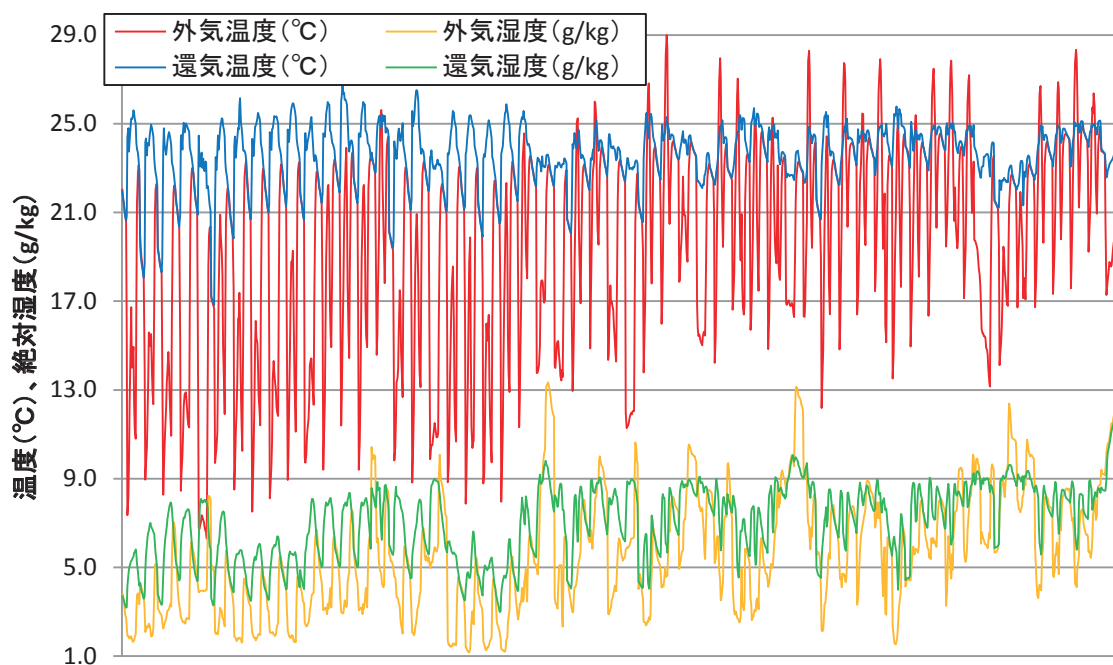


図 254 冬季期間中の外気温湿度、室内温湿度

表 98 冬季分析結果

	実測	計算	誤差
全熱能力期間積算値[kWh]	2473.5	2233.7	-9.69%
潜熱能力期間積算値[kWh]	960.5	—	—
消費電力期間積算値[kWh]	653.2	651.2	-0.31%
COP	3.79	—	—
COP (潜熱)	1.47	—	—

期間積算で全熱能力の誤差が 10%弱、消費電力量の誤差が 1%弱となった。

消費電力は計算値と実測値が期間積算値ではほぼ合致しているが、負荷率が高い条件においては実測値が計算値と比較して高い値を示すなど部分的に誤差が生じている。

3.5.7 圧縮機回転数を入力項としたヒートポンプデシカント調湿外調機特性式の精度検証、特性式作成

前項で示した特性式は入力項を室外温度、室外湿度、室内温度、室内湿度、潜熱能力としているためエネルギー計算に用いる際の利便性が高い。一方、前項で示した通り消費電力試算において若干の乖離が見られるため、圧縮機の回転数を入力項としたシステムの運転状態をダイレクトに表現できる特性式を用いた計算結果と実測結果との妥当性検証を行った。

3.5.7.1 消費電力計算

消費電力の計算フローを下記に示す。

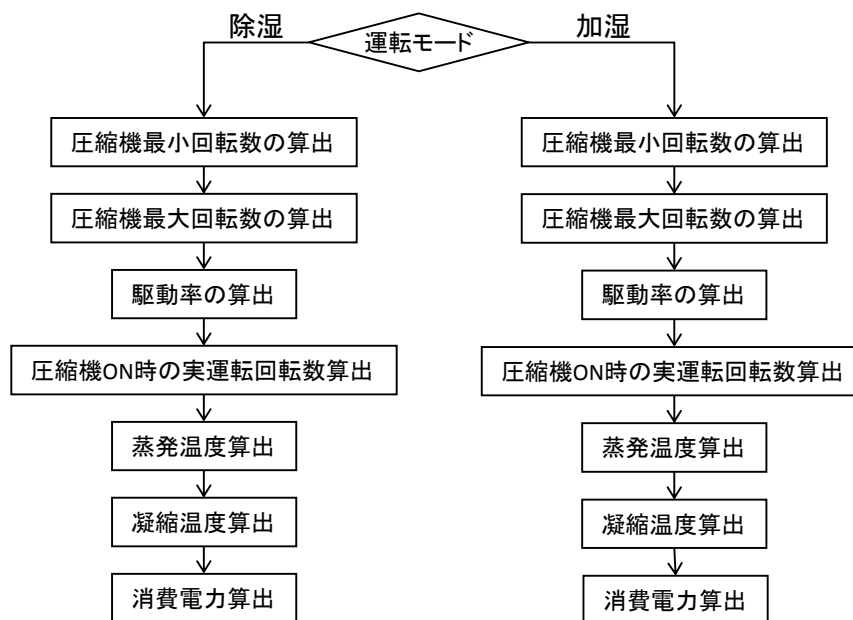


図 255 消費電力計算フロー

3.5.7.2 圧縮機最小回転数の計算

A ₁₁ ~A ₁₅	: 除湿運転時圧縮機最小回転数係数	
A ₂₁ ~A ₂₅	: 加湿運転時圧縮機最小回転数係数	
T _{OA}	: 外気温度	[°C]
X _{OA}	: 外気絶対湿度	[kg/kg]
T _{RA}	: 室内温度	[°C]
X _{RA}	: 室内絶対湿度	[kg/kg]
RS _{min}	: 圧縮機最小回転数	[rps]

表 99 圧縮機最小回転数係数

運転モード	A ₁	A ₂	A ₃	A ₄	A ₅
除湿	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅
加湿	A ₂₁	A ₂₂	A ₂₃	A ₂₄	A ₂₅

$$RS_{min} = A_1 + A_2 \times T_{OA} + A_3 \times X_{OA} + A_4 \times T_{RA} + A_5 \times X_{RA}$$

式 173

3.5.7.3 圧縮機最大回転数の計算

B ₁₁ ~B ₁₅	: 除湿運転時圧縮機最大回転数係数	
B ₂₁ ~B ₂₅	: 加湿運転時圧縮機最大回転数係数	
T _{OA}	: 外気温度	[°C]
X _{OA}	: 外気絶対湿度	[kg/kg]
T _{RA}	: 室内温度	[°C]
X _{RA}	: 室内絶対湿度	[kg/kg]
RS _{max}	: 圧縮機最大回転数	[rps]

表 100 圧縮機最大回転数係数

運転モード	B ₁	B ₂	B ₃	B ₄	B ₅
除湿	B ₁₁	B ₁₂	B ₁₃	B ₁₄	B ₁₅
加湿	B ₂₁	B ₂₂	B ₂₃	B ₂₄	B ₂₅

$$RS_{max} = B_1 + B_2 \times T_{OA} + B_3 \times X_{OA} + B_4 \times T_{RA} + B_5 \times X_{RA}$$

式 174

3.5.7.4 圧縮機駆動率の算出

CU	: 圧縮機駆動率	
RS _{min}	: 圧縮機最小回転数	[rps]
RS _{pre}	: 補正前圧縮機回転数	[rps]

表 101 圧縮機駆動率の算出

優先順位	条件	アクション
1	RS _{pre} < RS _{min}	CU = RS _{pre} / RS _{min}
2	Else	CU = 1.0

3.5.7.5 圧縮機 ON 時の実運転回転の算出

RS _{min}	: 圧縮機最小回転数	[rps]
RS _{min}	: 圧縮機最小回転数	[rps]
RS _{pre}	: 補正前圧縮機回転数	[rps]
RS	: 圧縮機回転数	[rps]

表 102 圧縮機回転数の算出

優先順位	条件	アクション
1	RS _{pre} < RS _{min}	RS=RS _{min}
2	RS _{max} < RS _{pre}	RS=RS _{max}
3	Else	RS=RS _{pre}

3.5.7.6 蒸発温度の算出

C ₁₁ ~C ₁₈	: 除湿運転時蒸発温度係数	
C ₂₁ ~C ₂₈	: 加湿運転時蒸発温度係数	
T _{OA}	: 外気温度	[°C]
X _{OA}	: 外気絶対湿度	[kg/kg]
T _{RA}	: 室内温度	[°C]
X _{RA}	: 室内絶対湿度	[kg/kg]
V _{OA}	: 給気風量	[m ³ /h] *JIS 風量測定条件 (20°C) の体積風量
V _{RA}	: 排気風量	[m ³ /h] *JIS 風量測定条件 (20°C) の体積風量
RS	: 圧縮機回転数	[rps]
Te	: 蒸発温度	[°C]

表 103 蒸発温度係数

運転モード	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈
除湿	C ₁₁	C ₁₂	C ₁₃	C ₁₄	C ₁₅	C ₁₆	C ₁₇	C ₁₈
加湿	C ₂₁	C ₂₂	C ₂₃	C ₂₄	C ₂₅	C ₂₆	C ₂₇	C ₂₈

$$Te = C_1 + C_2 \times T_{OA} + C_3 \times X_{OA} + C_4 \times T_{RA} + C_5 \times X_{RA} + C_6 \times V_{OA} + C_7 \times V_{RA} + C_8 \times RS$$

式 175

3.5.7.7 凝縮温度の算出

D ₁₁ ~D ₁₈	: 除湿運転時凝縮温度係数	
D ₂₁ ~D ₂₈	: 加湿運転時凝縮温度係数	
T _{OA}	: 外気温度	[°C]
X _{OA}	: 外気絶対湿度	[kg/kg]
T _{RA}	: 室内温度	[°C]
X _{RA}	: 室内絶対湿度	[kg/kg]
V _{OA}	: 給気風量	[m ³ /h] *JIS 風量測定条件 (20°C) の体積風量
V _{RA}	: 排気風量	[m ³ /h] *JIS 風量測定条件 (20°C) の体積風量
RS	: 圧縮機回転数	[rps]
Tc	: 凝縮温度	[°C]

表 104 凝縮温度係数

運転モード	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₈
除湿	D ₁₁	D ₁₂	D ₁₃	D ₁₄	D ₁₅	D ₁₆	D ₁₇	D ₁₈
加湿	D ₂₁	D ₂₂	D ₂₃	D ₂₄	D ₂₅	D ₂₆	D ₂₇	D ₂₈

$$T_c = D_1 + D_2 \times T_{OA} + D_3 \times X_{OA} + D_4 \times T_{RA} + D_5 \times X_{RA} + D_6 \times V_{OA} + D_7 \times V_{RA} + D_8 \times RS$$

式 176

3.5.7.8 消費電力の算出

E ₁₁ ~E ₁₄	: 除湿消費電力係数	
E ₂₁ ~E ₂₄	: 加湿消費電力係数	
F ₁ ~F ₃	: 圧縮機消費電力補正係数	
ON _{rate}	: リモコン ON 率	
CU	: 圧縮機駆動率	
RS	: 圧縮機回転数	[rps]
Te	: 蒸発温度	[°C]
Tc	: 凝縮温度	[°C]
W _{base}	: ベース消費電力	[W]
W _{fan}	: ファン消費電力	[W]
W _{comp}	: 圧縮機消費電力	[W]
W	: 消費電力	[W]

表 105 消費電力係数

運転モード	E ₁	E ₂	E ₃	E ₄
除湿	E ₁₁	E ₁₂	E ₁₃	E ₁₄
加湿	E ₂₁	E ₂₂	E ₂₃	E ₂₄

$$W_{comp} = E_1 \times RS^2 + E_2 \times T_e^2 + E_3 \times T_c^2 + E_4$$

式 177

表 106 圧縮機消費電力補正

優先順位	条件	アクション
1	運転モード=加湿	$W_{comp} = F_1 \times W_{comp}^2 + F_2 \times W_{comp} + F_3$
2	Else	$W_{comp} = W_{comp}$

$$W = ON_{rate} \times (CU \times (W_{comp} + W_{fan} + W_{base}) + (1 - CU) \times (W_{fan} + W_{base})) + (1 - ON_{rate}) \times W_{base}$$

式 178

3.5.7.9 給気温湿度の算出

給気温度の計算フローを下記に示す。

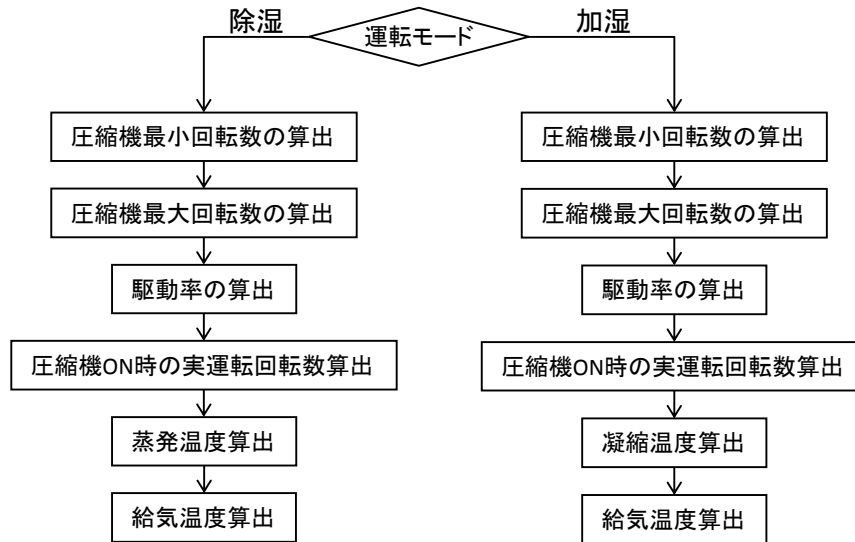


図 256 給気温度計算フロー

給気湿度の計算フローを下記に示す。

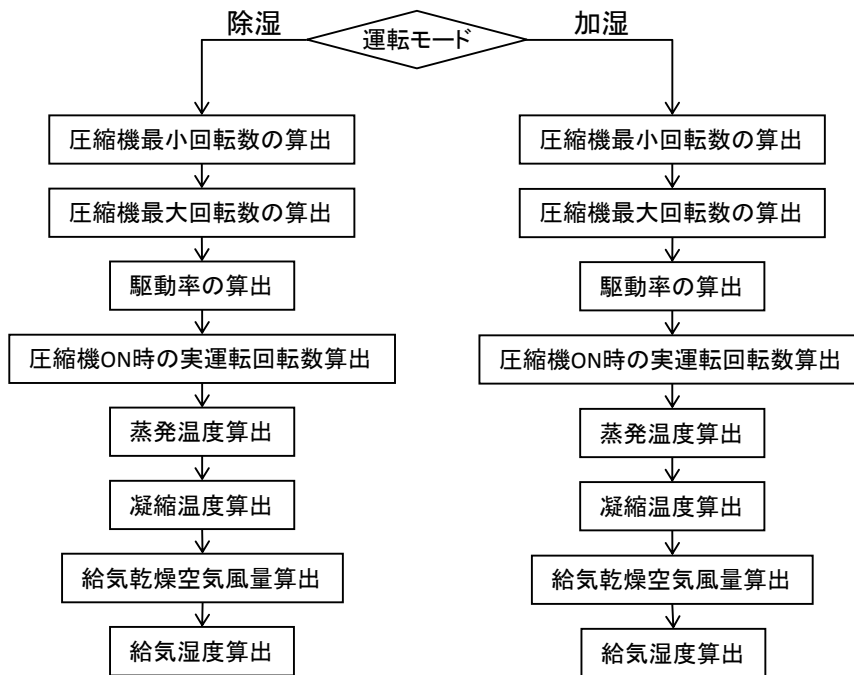


図 257 給気湿度計算フロー

3.5.7.10 給気温度算出

$G_{11}\sim G_{17}$: 除湿給気温度係数	
$G_{21}\sim G_{27}$: 加湿給気温度係数	
CU	: 圧縮機駆動率	
RS	: 圧縮機回転数	[rps]
V_s	: 給気風量	[kg/sec]
V_e	: 排気風量	[kg/sec]
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
T_e	: 蒸発温度	[°C]
T_c	: 凝縮温度	[°C]
T_{SA}	: 給気温度	[°C]

表 107 給気温度の算出

優先順位	条件	アクション
1	運転モード =加湿	$T_{SA}' = T_{oa} + G_{21} \times RS + G_{22} + RS^2 + G_{23} \times T_{OA}$ $+ G_{24} \times X_{OA} + G_{25} \times T_{RA} + G_{26} \times T_c + G_{27}$
2	運転モード =除湿	$T_{SA}' = T_{oa} + G_{11} \times RS + G_{12} + RS^2 + G_{13} \times T_{OA}$ $+ G_{14} \times X_{OA} + G_{15} \times T_{RA} + G_{16} \times T_e + G_{17}$
3	Else	T_{OA}

$$T_{SA} = CU \times T_{SA}' + (1 - CU) \times T_{OA}$$

式 179

3.5.7.11 給気湿度の算出

$H_{11}\sim H_{16}$: 除湿給気湿度係数	
$H_{21}\sim H_{27}$: 加湿給気湿度係数	
CU	: 圧縮機駆動率	
RS	: 圧縮機回転数	[rps]
V_{sDA}	: 給気乾燥空気風量	[kg/sec]
T_c	: 凝縮温度	[°C]
T_e	: 蒸発温度	[°C]
T_{OA}	: 外気温度	[°C]
T_{RA}	: 室内温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
X_{RA}	: 室内絶対湿度	[kg/kg]
X_{SA}	: 給気湿度	[kg/kg]

表 108 給気湿度の算出

優先順位	条件	アクション
1	運転モード =除湿	$X_{SA}' = X_{OA} - (H_{11} + H_{12} \times T_c + H_{13} \times T_e + H_{14} \times X_{RA} + H_{15} \times RS + H_{16} \times X_{OA}) / V_{sDA}$
2	運転モード =加湿	$X_{SA}' = X_{OA} + (H_{21} + H_{22} \times T_c + H_{23} \times T_e + H_{24} \times X_{RA} + H_{25} \times RS + H_{26} \times RS^{0.5} + H_{27} \times T_{OA}) / V_{sDA}$
3	Else	$X_{SA}' = X_{OA}$

$$X_{SA} = CU \times X_{SA}' + (1 - CU) \times X_{SA}'$$

式 180

3.5.8 実測データとの比較検証

3.5.8.1 東京 A ビル（夏季）

東京 A ビルの実測結果を用いて、6/2～9/11 の期間におけるヒートポンプデシカント調湿外調機の圧縮機回転数を入力項とした性能特性式から計算される消費電力と実測値の比較を行った。比較は、一時間平均の値で行った。

図 258、表 109、図 259 に消費電力の実測値と性能特性式から計算した値の比較を示す。

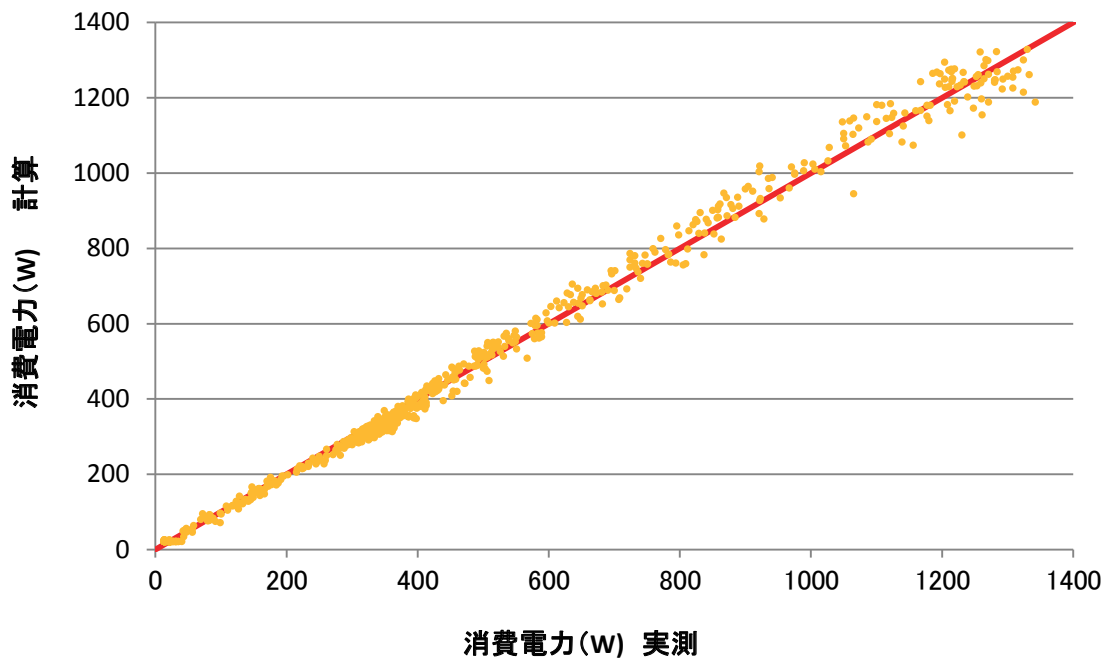


図 258 夏季消費電力比較（実測,計算）

表 109 夏季通期評価結果

	消費電力量 [Wh]	平均値	二乗平均 平方根誤差	平均 誤差	二乗平均 平方根誤差
実測	452855	381.7	—	—	—
計算	451399	—	19.5	0.3%	5.1%

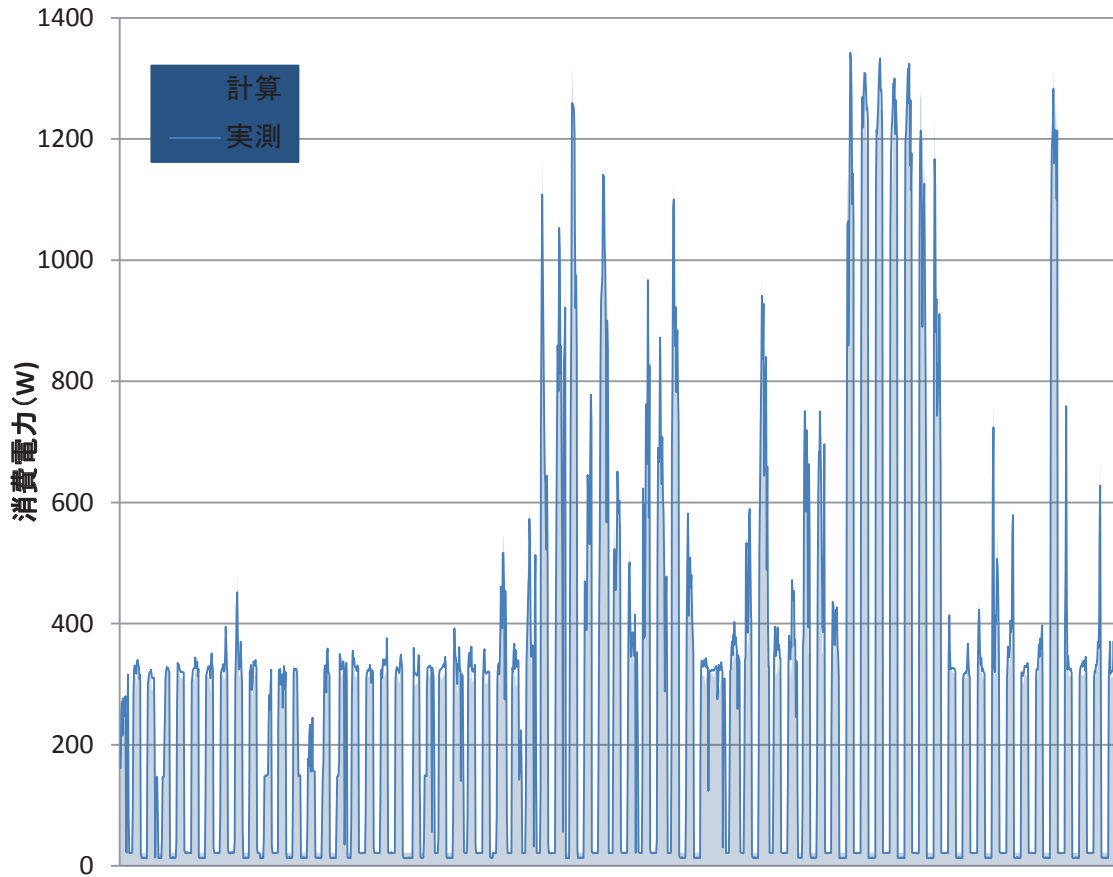


図 259 夏季消費電力比較(実測,計算)

期間積算で消費電力量の誤差が1%弱となった。

二乗平均平方根誤差で比較しても5%強の誤差となっており、入力項に圧縮機回転数を取ることによって精度よく消費電力が予測可能となった。

3.5.8.2 東京 A ビル（冬季）

冬期においては、1/30～4/18 期間にて分析を行った。夏季と同様に比較は、一時間平均の値で行った。

図 260、図 261、表 110 に消費電力の実測値と性能特性式から計算した値の比較を示す。

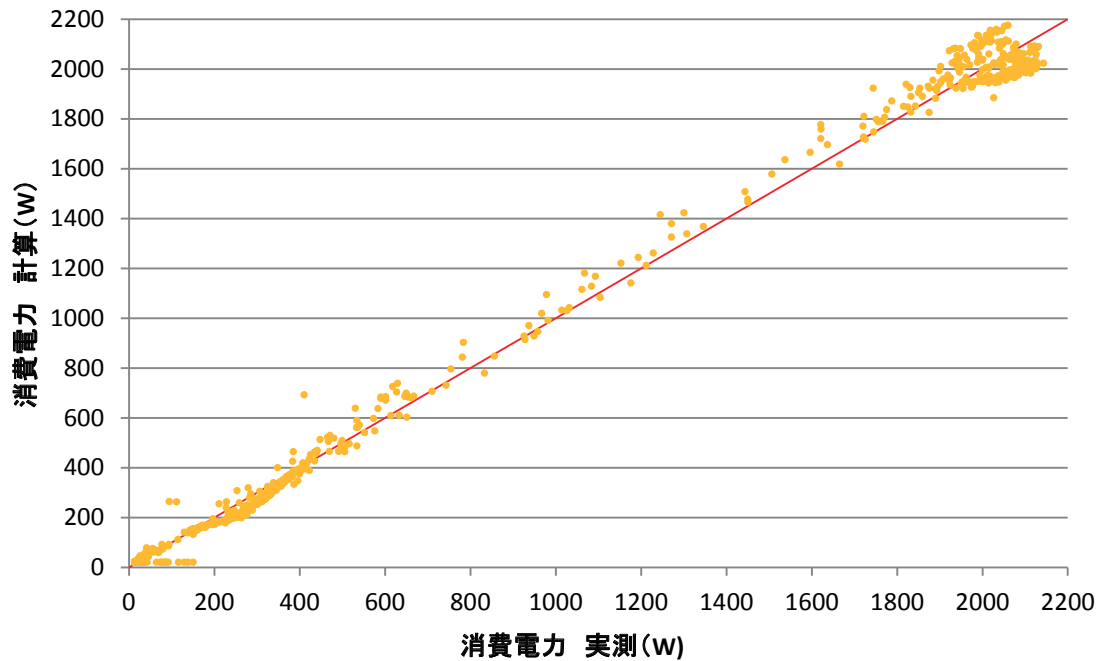


図 260 冬季消費電力比較（実測,計算）

表 110 冬季通期評価結果

	消費電力量 [Wh]	平均値	二乗平均 平方根誤差	平均 誤差	二乗平均 平方根誤差
実測	717483	490.1	—	—	—
計算	714789	—	39.8	0.4%	8.13%

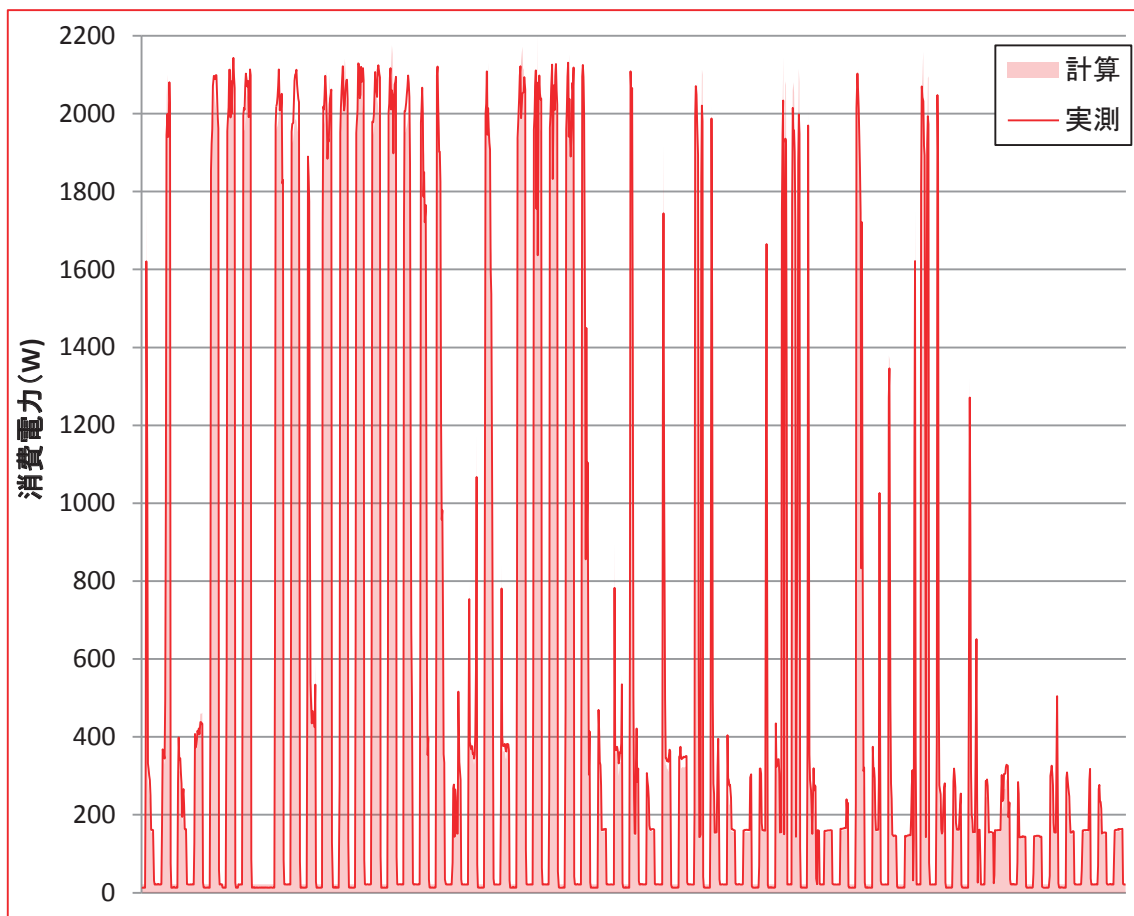


図 261 冬季消費電力比較(実測,計算)

期間積算で消費電力量の誤差が 1%弱となった。

二乗平均平方根誤差で比較しても 8%強の誤差となっており、入力項に圧縮機回転数を取ることによって精度よく消費電力が予測可能となった。

3.5.8.3 建築研究所（冬季）

建築研究所において実測した結果についても、実測値と計算値の比較を行った。2016/1/29～2016/2/7の期間のデータを用いて分析を行った。

消費電力の実測値と性能特性式から計算した値の比較を図 262、表 111、図 263 に示す。

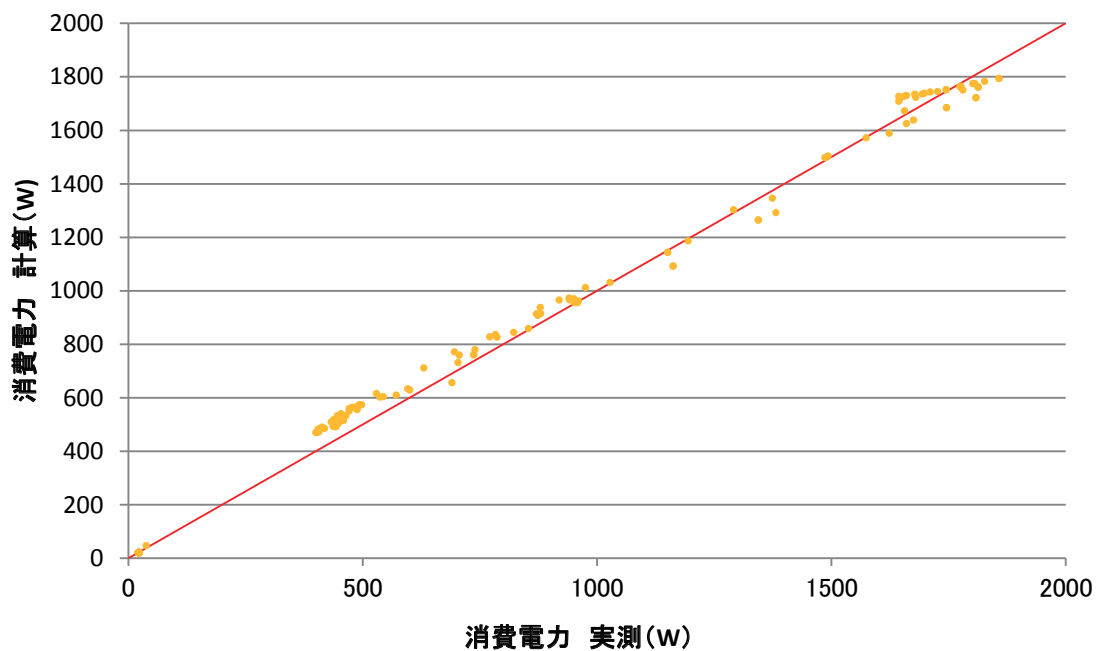


図 262 冬季消費電力比較(実測,計算)

表 111 冬季通期評価結果

	消費電力量 [Wh]	平均値	二乗平均 平方根誤差	平均 誤差	二乗平均 平方根誤差
実測	121821	507.6	—	—	—
計算	125687	—	41.4	3.2%	8.2%

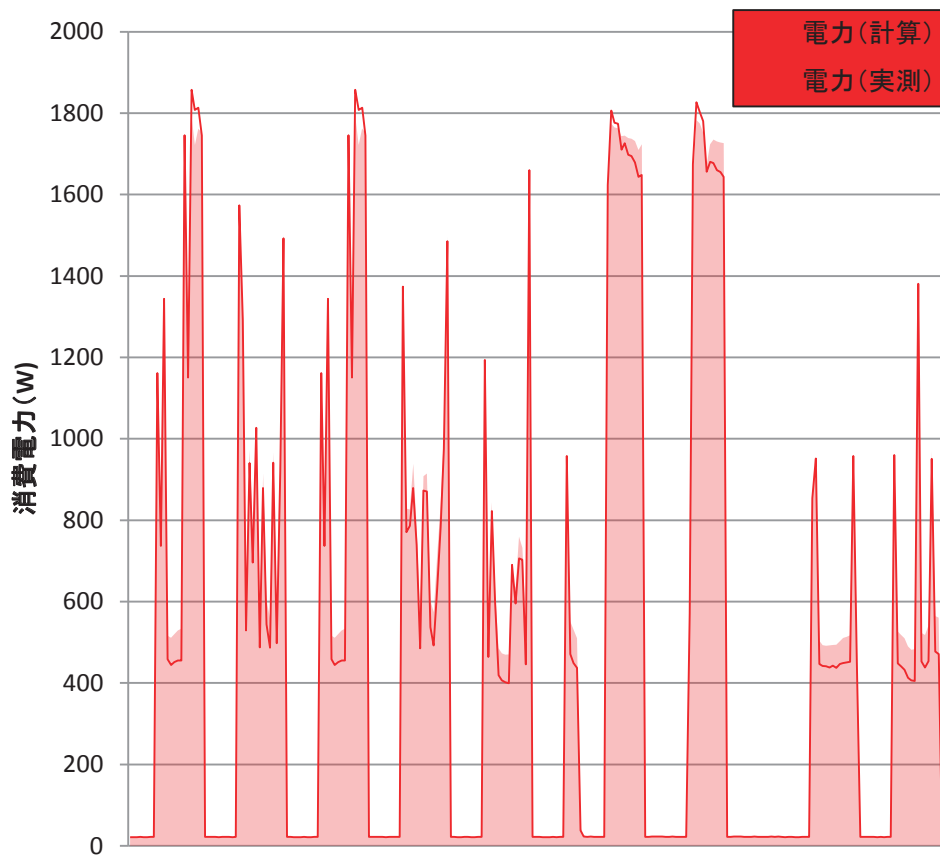


図 263 冬季消費電力比較(実測,計算)

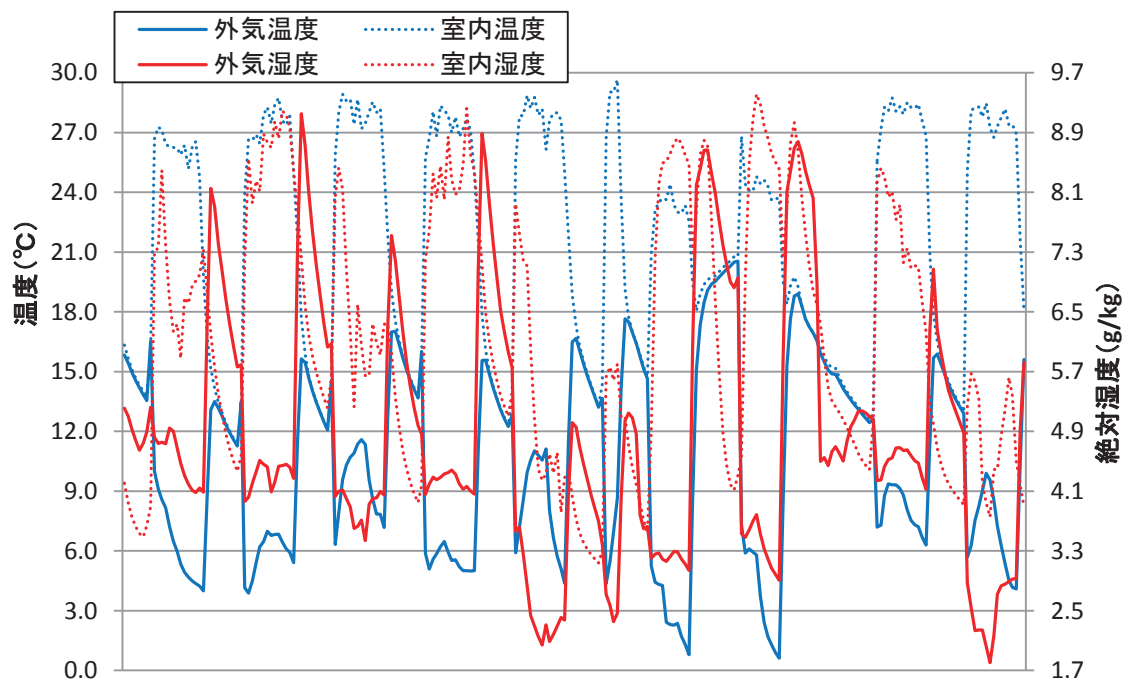


図 264 冬季期間中の外気温湿度、室内温湿度

期間積算で消費電力量の誤差が3%強、二乗平均平方根誤差で8%強の誤差となった。
 Aビルの結果よりも、若干大きめの誤差となったものの計算モデルの高い精度が確認できた。

3.5.9 省エネルギー基準への反映案（全熱計算法への反映）

現在の省エネルギー基準においては、空調負荷を潜熱、顕熱分離して取り扱わず全熱として取り扱っている。本項では、全熱負荷計算における調湿外調機の計算方法について提案を行う。

<入力>

L_{total}	: 室全熱負荷	[W]
LL_{out}	: 外気潜熱負荷	[W]
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]

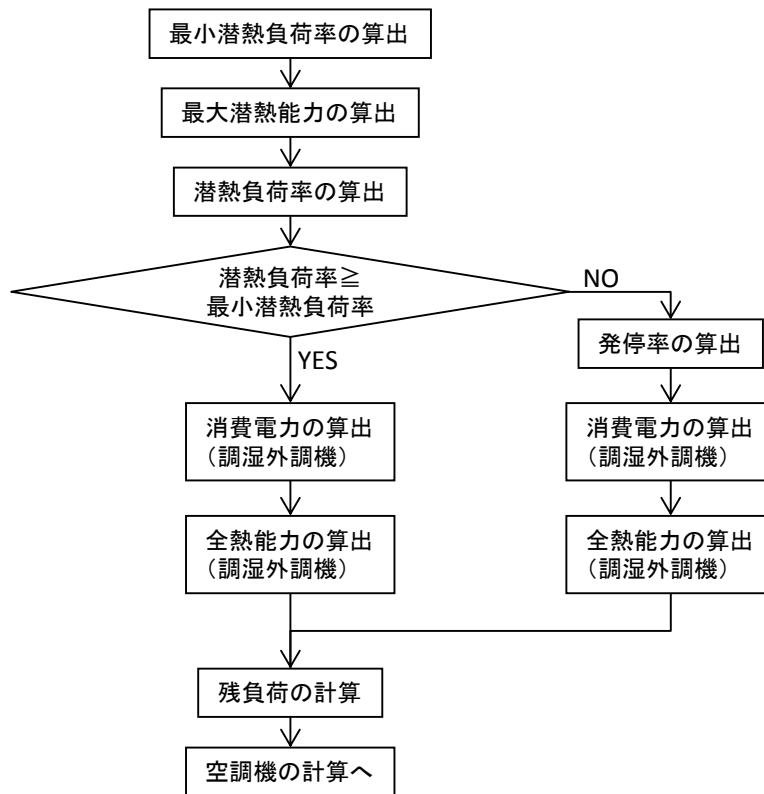


図 265 計算フロー

3.5.9.1 最小潜熱負荷率の算出

$\beta_{00} \sim \beta_{24}$: 最小潜熱負荷率係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
$PLRL_{min}$: 最小潜熱負荷率	

$$\text{最小潜熱負荷率} = \beta_{00} + \beta_{11} \times T_{OA} + \beta_{12} \times X_{OA} + \beta_{13} \times T_{RA} + \beta_{14} \times X_{RA} + \beta_{21} \times T_{OA}^2 + \beta_{22} \times X_{OA}^2 + \beta_{23} \times T_{RA}^2 + \beta_{24} \times X_{RA}^2$$

式 181

3.5.9.2 最大潜熱能力の算出

$\alpha_{00} \sim \alpha_{14}$: 最大潜熱能力係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
LC_{max}	: 最大潜熱能力	[W]

$$\text{最大潜熱能力 (LC}_{max}) = \alpha_{00} + \alpha_{11} \times T_{OA} + \alpha_{12} \times X_{OA} + \alpha_{13} \times T_{RA} + \alpha_{14} \times X_{RA}$$

式 182

3.5.9.3 潜熱負荷率の算出

LC_{max}	: 最大潜熱能力	[W]
LC	: 潜熱能力(負荷)	[W]
$PLRL$: 潜熱負荷率	

$$\text{潜熱負荷率 (PLRL)} = \text{潜熱負荷 (LC)} / \text{最大潜熱能力 (LC}_{max})$$

式 183

但し、潜熱負荷率は最小潜熱負荷率以上、1以下で制限する。

3.5.9.4 発停率の算出

潜熱負荷率が最小潜熱負荷率を下回る場合、発停率の算出を行う。

O_{fre}	: 発停率	
$PLRL_{min}$: 最小潜熱負荷率	
LC	: 潜熱能力 (負荷)	[W]
LC_{max}	: 最大潜熱能力	[W]

$$\text{発停率 (O}_{fre}) = LC / (PLRL_{min} \times LC_{max})$$

式 184

3.5.9.5 消費電力の算出

$\delta_{00} \sim \delta_{34}$: 消費電力係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
PLRL	: 潜熱負荷率<最小潜熱負荷率以上、1以下の範囲に制限される>	
O_{fre}	: 発停率	
W_{fan}	: ファン消費電力	[W]: 定格静圧, 定格風量における消費電力を用いる

$$\text{消費電力} = (\delta_{00} + \delta_{11} \times T_{OA} + \delta_{12} \times X_{OA} + \delta_{13} \times T_{RA} + \delta_{14} \times X_{RA} + \delta_{15} \times \text{PLRL} + \delta_{21} \times T_{OA} \times \text{PLRL} + \delta_{22} \times X_{OA} \times \text{PLRL} + \delta_{23} \times T_{RA} \times \text{PLRL} + \delta_{24} \times X_{RA} \times \text{PLRL} + \delta_{31} \times T_{OA} \times \text{PLRL}^2 + \delta_{32} \times X_{OA} \times \text{PLRL}^2 + \delta_{33} \times T_{RA} \times \text{PLRL}^2 + \delta_{34} \times X_{RA} \times \text{PLRL}^2) \times O_{fre} + W_{fan} \times (1 - O_{fre})$$

式 185

3.5.9.6 全熱能力の算出

$\gamma_{00} \sim \gamma_{34}$: 全熱能力係数	
T_{OA}	: 外気温度	[°C]
X_{OA}	: 外気絶対湿度	[kg/kg]
T_{RA}	: 室内温度	[°C]
X_{RA}	: 室内絶対湿度	[kg/kg]
PLRL	: 潜熱負荷率<最小潜熱負荷率以上、1以下の範囲に制限される>	
O_{fre}	: 発停率	

$$\text{全熱能力} = (\gamma_{00} + \gamma_{11} \times T_{OA} + \gamma_{12} \times X_{OA} + \gamma_{13} \times T_{RA} + \gamma_{14} \times X_{RA} + \gamma_{15} \times \text{PLRL} + \gamma_{21} \times T_{OA} \times \text{PLRL} + \alpha_{22} \times X_{OA} \times \text{PLRL} + \alpha_{23} \times T_{RA} \times \text{PLRL} + \alpha_{24} \times X_{RA} \times \text{PLRL} + \alpha_{31} \times T_{OA} \times \text{PLRL}^2 + \alpha_{32} \times X_{OA} \times \text{PLRL}^2 + \alpha_{33} \times T_{RA} \times \text{PLRL}^2 + \alpha_{34} \times X_{RA} \times \text{PLRL}^2) \times O_{fre}$$

式 186

3.5.9.7 残負荷の算出

L_{total}	: 室全熱負荷	[W]
C_{total}	: 全熱能力	[W]
L_{rem}	: 残負荷	[W]

$$L_{rem} (\text{残負荷}) = L_{total} - C_{total}$$

式 187

3.5.10 ヒートポンプデシカント調湿外調機の性能計測方法

ヒートポンプデシカント調湿外調機の性能を計測する方法についての考え方を整理した。本計測手法に基づいて技術資料の表記データは作成されている。

3.5.10.1 性能評価を行う上での評価項目・計測器必要精度

定常状態で運転する場合の測定項目および測定項目に使用する計測器の必要精度はJISB8615に準ずる。JIS改訂時には、改訂版の精度に改訂する。

表 112 非定常計測における測定項目と必要精度

	測定項目	必要精度
温 度	還気 (RA) 乾球温度	定常時 能力用：±0.1℃ 性能用：±0.2℃ 非定常応答性 10秒以内 0.2℃未満到達のこと
	外気 (OA) 乾球温度	
	給気 (SA) 乾球温度	
	排気 (EA) 乾球温度	
	還気 (RA) 湿球温度	定常時 能力用：±0.1℃ 性能用：±0.2℃ 非定常応答性 10秒以内 0.2℃未満到達のこと
	外気 (OA) 湿球温度	
	給気 (SA) 湿球温度	
	排気 (EA) 湿球温度	
湿 度	還気 (RA) 相対湿度	定常時 能力用：±2%Rh 性能用：±3%Rh 非定常応答性 30秒以内に变化量の90%応答
	外気 (OA) 相対湿度	
	給気 (SA) 相対湿度	
	排気 (EA) 相対湿度	
	還気 (RA) 絶対湿度	定常時 能力用：±1g/kg-DA 性能用：±1g/kg-DA 非定常応答性 30秒以内に变化量の90%応答
	外気 (OA) 絶対湿度	
	給気 (SA) 絶対湿度	
	排気 (EA) 絶対湿度	

湿度測定は湿球に頼らず、精度信頼における絶対湿度センサーまたは相対湿度センサーの使用を可とする。

3.5.10.2 性能評価時の計測方法

性能評価を実施する際の試験機の設置方法を下図に示す。

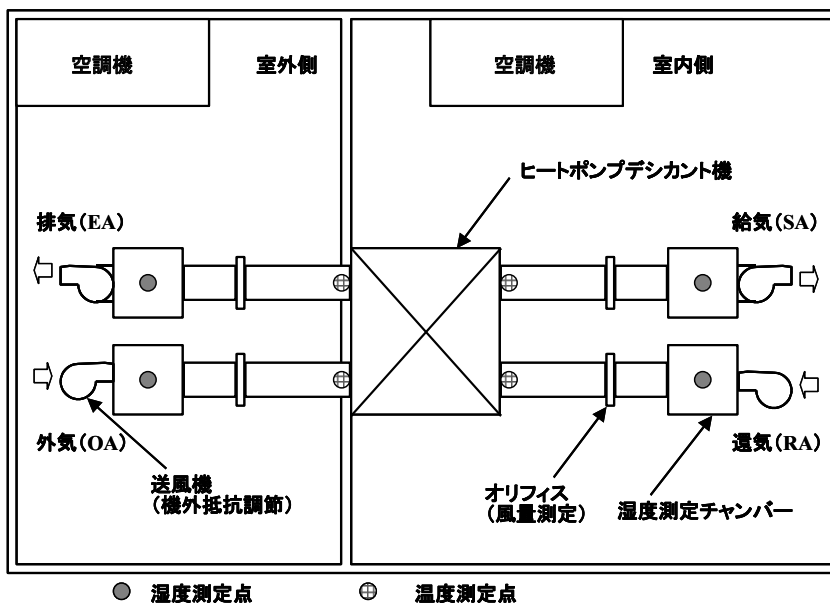


図 266 試験機の設置方法

- 送風機が被試験機に内蔵されており、且つ機外静圧を所定の値に調整可能な場合は、外部ファンは不要とする
- 定格機外静圧を表示したものにあってはその時の機外静圧を加える
- 温度・湿度の測定点までの管路は、必要に応じて断熱すること
- 温度・湿度の測定点は、管路内の平均的な値が得られるよう位置調節や混合部設置を行うこと、またそれらは風量測定部と一体としてもよい
- 各室の外壁と仕切り板には十分な断熱を施すこと
- 各室の温度・湿度の分布は、均一となるよう留意すること

また、計測時の室内、室外空気条件は、表 113 の範囲に入っていること

表 113 計測時における室内、室外空気条件

室内		室外	
乾球温度	湿球温度	乾球温度	湿球温度
設定温度 $\pm 0.3^{\circ}\text{C}$	設定温度 $\pm 0.2^{\circ}\text{C}$	設定温度 $\pm 0.3^{\circ}\text{C}$	設定温度 $\pm 0.2^{\circ}\text{C}$

適用機種がバッチ式である場合、加湿能力の測定は次のように行う。サイクルはバッチ運転において各バッチを1周りすることを表す。また平均値は計測値から求めた能力値等の積分平均値とする。

- ・ 室外側、室内側の空気条件が表 113 の条件で安定した後、被試験機を運転する
- ・ 試験条件が安定したことを確認し1時間以上運転した後に、連続3サイクル計測し平均値データを採用、記録する。このとき算出した能力のバラツキがない（各サイクルの最大・最小値が平均値の $\pm 3\%$ 以内にある）ことを確認すること
- ・ 温調室の試験条件変動などで被試験機が安定しない場合は連続6サイクル測定し、全て上記条件を満足できない場合は、最もバラツキの小さい連続3サイクルの平均値を採用する。そのとき還気、外気、給気の乾球温度、湿球温度（相対湿度）の変動も記録すること

風量、静圧の計測方法は JISB8628 に準ずるものとする。

3.5.11 機種選定手順

カタログに記載されている最大能力表を用いたヒートポンプデシカント外調機および高顕熱形ビル用マルチエアコンの機種選定手順について整理した。

3.5.11.1 ヒートポンプデシカント調湿外調機選定の手順（フローチャート）

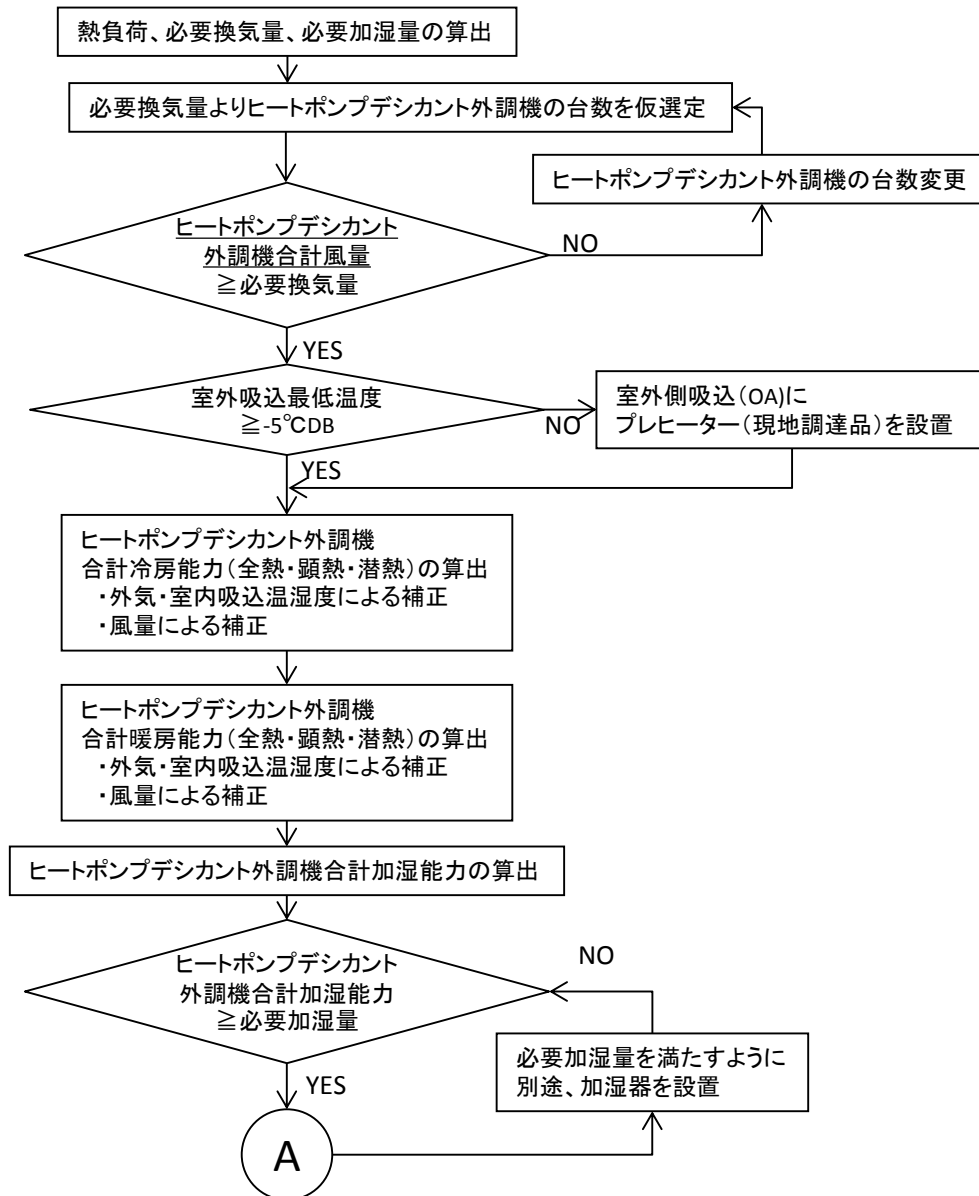


図 267 ヒートポンプデシカント調湿外調機選定時のフローチャート

3.5.11.2 ヒートポンプデシカント調湿外調機の風量

「風量一定機能を使用する場合（工場出荷時の設定）」

500m³/h（規定風量）：規定の機外静圧範囲内（50～220Pa）において自動で風量を調整

「風量一定機能を使用しない場合（現地設定）」

450～550m³/h（規定の風量範囲内）：規定機外静圧範囲内において、現地設定にて給気・排気ファンステップを固定し、手動で風量を調整

3.5.11.3 ヒートポンプデシカント調湿外調機の冷房能力算出方法

(1) 冷房能力（全熱・顕熱）を算出する

冷房能力（全熱・顕熱）

＝除湿冷房性能特性表より外気・室内吸込温湿度を補正した除湿冷房能力（全熱・顕熱）
 ×風量による除湿冷房能力補正係数

式 188

(2) 冷房能力（潜熱）を算出する

冷房能力（潜熱）＝冷房能力（全熱）－冷房能力（顕熱）

式 189

3.5.11.4 調湿外調機の暖房能力算出方法

(1) 暖房能力（全熱・顕熱）を算出する。

暖房能力（全熱・顕熱）

＝加湿暖房性能特性表より外気・室内吸込温湿度を補正した加湿暖房能力（全熱・顕熱）
 ×風量による加湿暖房能力補正係数

式 190

(2) 暖房能力（潜熱）を算出する

暖房能力（潜熱）＝暖房能力（全熱）－暖房能力（顕熱）

式 191

3.5.11.5 調湿外調機の加湿能力算出方法

加湿能力[kg/h]＝暖房能力（潜熱）×3600[sec]÷2500[kJ/kg]

式 192

3.5.11.6 風量による能力の補正

表 114 除湿冷房能力補正係数

		給気		
		550m ³ /h	500m ³ /h	450m ³ /h
排 気	550m ³ /h	1.05	1.03	1.01
	500m ³ /h	1.02	1.00	0.98
	450m ³ /h	0.99	0.97	0.95

表 115 加湿暖房能力補正係数

		給気		
		550m ³ /h	500m ³ /h	450m ³ /h
排 気	550m ³ /h	1.06	1.04	1.02
	500m ³ /h	1.02	1.00	0.98
	450m ³ /h	0.98	0.96	0.94

3.5.11.7 ビル用マルチエアコンの選定の手順

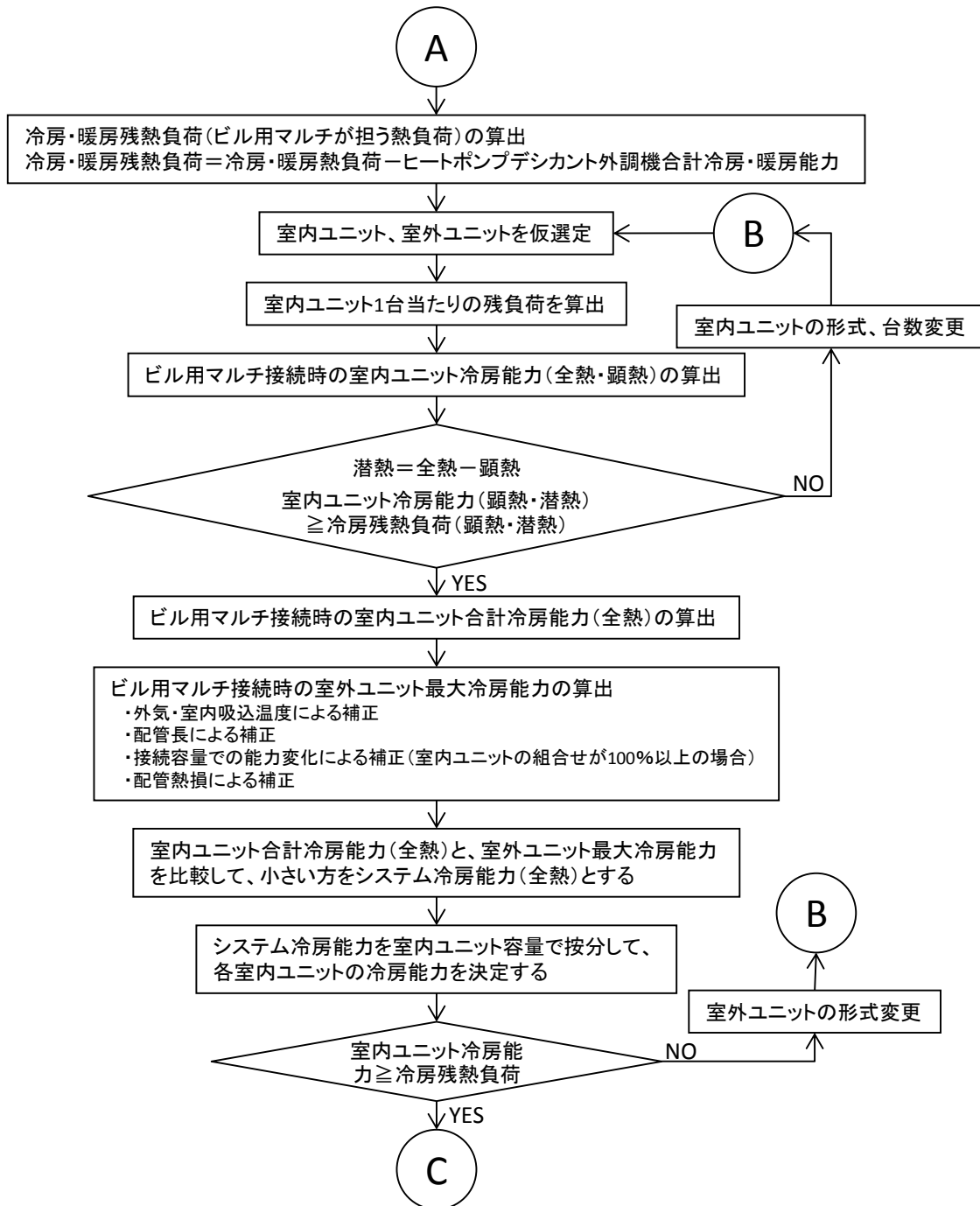


図 268 ビル用マルチエアコン選定時のフローチャート

3.5.11.8 残熱負荷算出方法

(1) 冷房残熱負荷（顕熱・潜熱）を算出する

$$\text{冷房残熱負荷（顕熱・潜熱）} = \text{冷房熱負荷（顕熱・潜熱）} - \text{DESICA 合計冷房能力（顕熱・潜熱）}$$

式 193

(2) 暖房残熱負荷（顕熱）を算出する

$$\text{暖房残熱負荷（顕熱）} = \text{暖房熱負荷（顕熱）} - \text{DESICA 合計暖房能力（顕熱）}$$

式 194

3.5.11.9 ビル用マルチ接続時の室内ユニット冷房能力（全熱・顕熱）を算出する

$$\begin{aligned} & \text{ビル用マルチ接続時の室内ユニット（冷房能力）} \\ & = \text{ビル用マルチ接続時の室内ユニット定格冷房能力} \times \text{室内吸込温度による能力補正係数} \end{aligned}$$

式 195

3.5.11.10 ビル用マルチ接続時の室内ユニット冷房能力（潜熱）を算出する

$$\begin{aligned} & \text{ビル用マルチ接続時の室内ユニット冷房能力（潜熱）} \\ & = \text{ビル用マルチ接続時の室内ユニット冷房能力（全熱）} \\ & \quad - \text{ビル用マルチ接続時の室内ユニット冷房能力（顕熱）} \end{aligned}$$

式 196

3.5.11.11 ビル用マルチ室外ユニット最大冷房能力を算出する

$$\begin{aligned} & \text{ビル用マルチ室外ユニット最大冷房能力} \\ & = \text{ビル用マルチ室外ユニット定格冷房能力} \\ & \quad \times \text{外気・室内吸込温度による能力補正係数} \\ & \quad \times \text{能力特性図より求めたその接続容量での能力変化率} \\ & \quad \times \text{配管帳による能力補正係数} \\ & \quad \times \text{配管熱損による能力補正係数} \end{aligned}$$

式 197

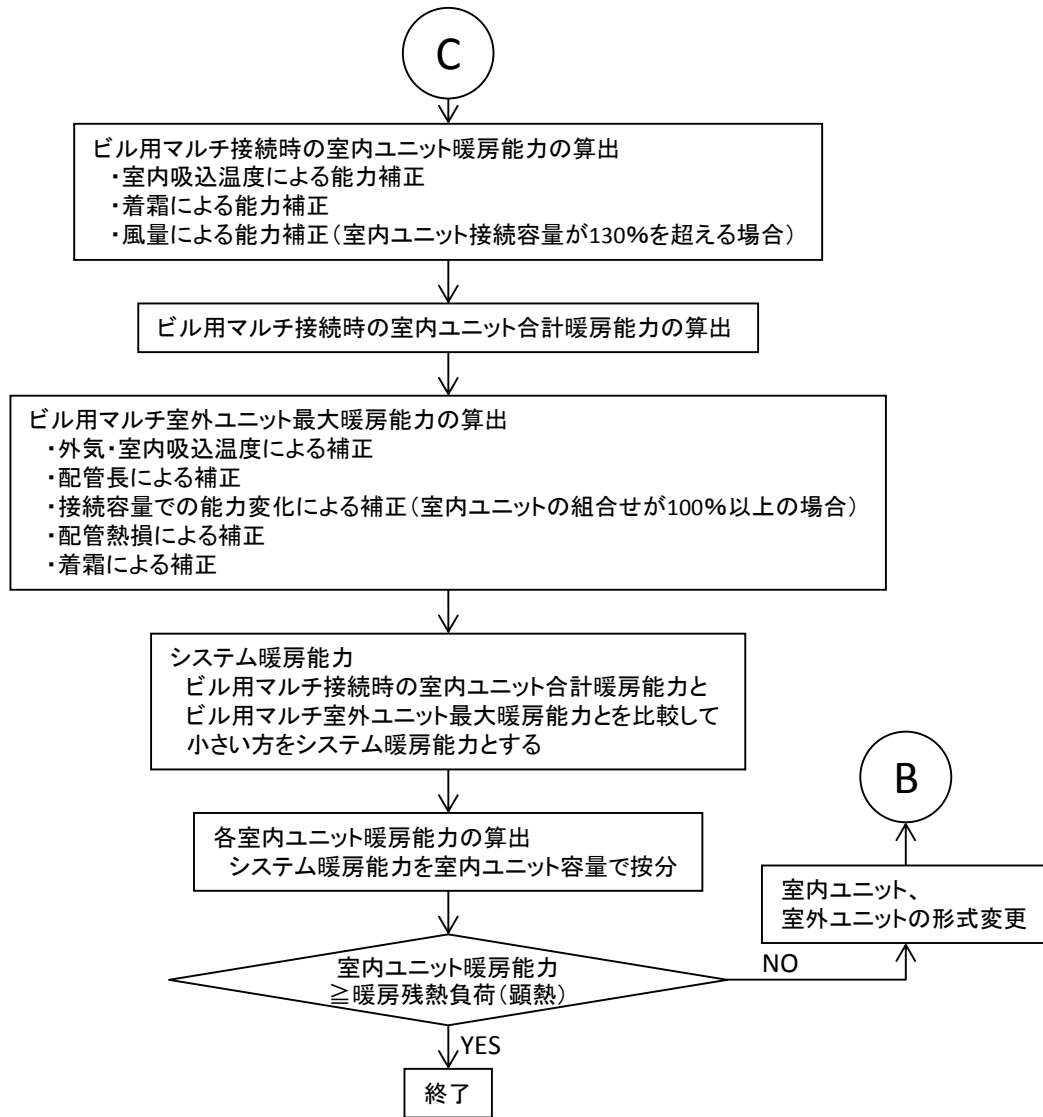


図 269 ビル用マルチエアコン選定時のフローチャート

3.5.11.12 残熱負荷算出方法

(1) 室内ユニット接続容量が 130%以下の場合

ビル用マルチ接続時の室内ユニット暖房能力
＝ビル用マルチ接続時の室内ユニット定格暖房能力
×室内吸込温度による能力補正係数×着霜による能力補正係数

(2) 室内ユニット接続容量が 130%超の場合

ビル用マルチ接続時の室内ユニット暖房能力
＝ビル用マルチ接続時の室内ユニット定格暖房能力
×室内吸込温度による能力補正係数
×着霜による能力補正係数
×弱風量/強風量（急風量）

3.5.11.13 ビル用マルチ室外ユニット最大暖房能力を算出する

ビル用マルチ室外ユニット最大暖房能力
＝ビル用マルチ室外ユニット定格暖房能力
×外気・室内吸込温度による能力補正係数
×能力特性線図より求めたその接続容量での能力変化率
×配管長による能力補正係数
×配管熱損による能力補正係数
×着霜による能力補正係数

式 198

3.5.11.14 各室内ユニット暖房能力を算出する

(1) 室内ユニット接続容量が 130%以下の場合

室内ユニット暖房能力＝システム暖房能力×室内ユニット容量／室内ユニット容量の合計

式 199

(2) 室内ユニット接続容量が 130%超の場合

室内ユニット暖房能力
＝システム暖房能力×室内ユニット×弱風量/強風量（急風量）
／（室内ユニット容量×弱風量/強風量（急風量））の合計

式 200

4 住宅における居住者の通風・冷房行為に関する実態把握と行為モデルの構築

4.1 調査の目的

住宅のエネルギー消費は、建物や所有機器の仕様のみならず、生活行動や機器使用などの居住者行動により大きく影響される。居住者行為のうち夏期の採涼に関しては、状況に応じて居住者が通風とエアコンとを使い分けることが一般的である。このため、そうした使い分けを伴う通風・冷房行為を如何に再現するかが、冷房エネルギー消費の予測・評価を行う上で重要な課題となる。

住宅の省エネルギー基準^{xiii}においては、通風措置の有無によらず、室内温度が設定温度(起居時 27°C、就寝時 28°C)を超えた場合に冷房が使用されるものとしている。しかしながら、設定温度を冷房使用室温とする考え方は、連続運転を前提とするものであり、間欠運転かつ通風利用を基本とする住宅にはマッチしない可能性がある。

5.8 負荷計算における通風量一定の仮定について(文献 xiii pp.662 より一部抜粋して引用)

- 在室時に、通風を利用し(換気回数で 5 回/h、又は 20 回/h にあたる外気量を導入し)冷房を停止したままで室温を 27°C(就寝時 28°C)以下に保つことができる場合は、通風を利用し冷房運転を停止した条件で対象居室の温度等を計算する(冷房負荷は発生しない)。
- 在室時に、通風を利用し(換気回数で 5 回/h、又は 20 回/h にあたる外気量を導入し)冷房を停止したままでは室温が 27°C(就寝時 28°C)を上回る場合には、通風利用を停止し、冷房により対象居室の温度を 27°C(就寝時 28°C)、相対湿度を 60%に制御して、冷房顕熱負荷、冷房潜熱負荷の算定を行う。

そこで、本章では各種調査を通じて住宅における通風と冷房の使用実態に関するデータを収集・整理し、以下の検討を行った。

- WEB アンケート調査に基づく通風・冷房行為のパターンとその決定要因の整理(4.2 節)
関西地域を対象とした WEB アンケートを通じて居住者が選択した通風・冷房行為を生活時間ごとに聴取し、通風・冷房行為のパターンを整理した。さらに、季節、世帯属性、住宅建物、住居環境および周辺環境と通風・冷房行為との関係性を分析した。分析結果から、行為選択と関連性の強い要素を生活時間別に抽出し、要素による行為選択の相違を整理した。加えて、行為選択の相違を説明する上での要素の重要度を示した。
- フィールド実測調査に基づく窓・エアコン操作の生起条件の分析(4.3 節)
関西地域に住まう 11 世帯を対象として、外気温度、室内温度・相対湿度、エアコン・窓の状態に関するデータを収集した。収集したデータを各種統計的手法により分析し、エアコン発/停、窓開/閉に関する行為の生起要因を時間区分ごとに探究した。
- 通風・冷房行為に関するモデルの考案(4.4 節)
調査結果に基づいて、エアコンの on/off 操作および窓の open/close 操作を生起させるトリガーとその閾値を設定し、通風・冷房行為に関するモデルを考案した。

4.2 WEB アンケート調査に基づく通風・冷房行為のパターンとその決定要因の整理

4.2.1 関連研究の動向と本調査の目的

通風・冷房行為のモデリングについては、行為決定の要因や特定の要因をパラメータとして行為遷移が起こる閾値あるいは確率を実態調査結果から抽出することで試みられている^{xiv, xv, xvi, xvii, xviii, xix, xx, xxi, xxii}。ただし、国外の既往研究では冷房を必要としない地域を対象とするものが多く、通風行為のみを対象としたモデルが提案されている。最近では、外気温や室温といった環境要素だけでなく、在室パターン等の居住者の生活行動に関連する要素や住宅形式や換気設備の種類等の建物仕様に関連する要素をも考慮することや、さらには世帯差を表現することが試みられるようになった^{xxiii, xxiv}。

これまでに、筆者は実測調査結果に基づいて通風・冷房行為のモデル化に取り組んできた^{xxv, xxvi}。構築したモデルは、室温を主な入力パラメータとして、行為を確率的に与えるものである。しかし、既往研究により室温などの環境要素以外にも様々な要素が行為選択に作用することが明らかにされている^{xxvii, xxviii, xxix, xxx, xxxi, xxxii, xxxiii, xxxiv, xxxv, xxxvi}。具体的には、冷房の仕方や窓開放には居住者の価値観や意識以外にも物理的・社会的な外的要因が様々な形で影響を与えることが指摘されている^{xxxvii, xxxviii}。外的要因には、騒音、大気汚染、臭気といった周辺の物理環境、防犯やプライバシーなどがあり、これらの要因は窓開放を阻害すると考えられている^{xxxix, xl}。さらに、これらの要素は住宅の形式（集合/戸建）や仕様（伝統/新興）、住戸位置によって影響の程度が異なることが報告されている^{xli, xlii, xliii, xliv, xlv}。また、居住者のライフスタイルとの関連性も強く、例えば朝の時間帯に窓開放を行うなど、通風・冷房行為に対しては時間帯によって世帯ごとに異なる方針が存在することが示されている^{xlii, xliii}。

前段のように、これまでの研究により通風・冷房行為の選択は周辺環境（騒音、治安など）、建物条件（住宅形式、住戸位置など）、世帯特性（生活時間、価値観など）といった様々な要素が作用することが明らかにされた。しかし、これらの多岐にわたる要素を包括して扱い、行為選択との関連性を定量的に示した事例は少ない。そこで、本検討では、関西地域を対象としたWEB アンケートを通じて居住者が選択した通風・冷房行為を不在時・外出時を含めて生活時間ごとに聴取した結果から、統計的手法により行為選択と関連性の強い要素を生活時間別に抽出し、要素による行為選択の相違を整理する。さらに、行為選択の相違を説明する上での要素の重要度を明らかにする。

4.2.2 調査概要

調査概要を表 116 に示す。調査は、インターネット調査会社を通じて WEB アンケートにより実施した。対象は、近畿2府4県（大阪府、京都府、兵庫県、和歌山県、奈良県、滋賀県）に住まう2世代同居の3～5人世帯（夫婦とその子より成る世帯）とし、第1回から第3回まで同一の世帯に対してアンケートへの協力を依頼した。ただし、事前にスクリーニングを行い、該当する世帯を対象を限定した。

表 116 調査概要

方法	WEBアンケート調査
対象地域	近畿2府4県（大阪府、京都府、兵庫県、和歌山県、奈良県、滋賀県）
対象世帯	以下の項目を満たす世帯 <ul style="list-style-type: none"> ・2世代（夫婦とその子より成る）が同居する3～5人世帯 ・室内飼ペットがいない ・家族に要介護者がいない ・夫婦の年齢がいずれも60歳以下である ・居間と主寝室のそれぞれにエアコンが1台以上設置されている
対象居室	居間、主寝室（夫婦の寝室）
調査期間	2013/8/9～8/19（第1回）、9/13～9/17（第2回）、10/23～10/25（第3回）

設問は、居間と主寝室（夫婦の寝室）における回答日から過去1週間の通風・冷房行為に関する項目に加え、回答者の基本情報として世帯属性（居住地域、家族構成、年齢、職業、世帯年収）、入居中の住宅建物に関する情報（建築年、住宅形式、住戸位置）、住居環境（風通し、日当たり）、住宅周辺の環境（騒音、大気汚染、臭気、人通り、車通り、治安）に関する項目を設けた。設問項目の一覧を表 117 に示す。なお、設問項目は、通風・冷房行為に影響すると考えられ、かつ物理的計測や統計資料により客観的な情報が入手可能である要素を中心に設定した。また、通風・冷房行為の選択は、就寝時、朝起床直後などといった生活時間により異なることが予想されたため、居住者の生活時間を「夜間、別室で寝ており、対象居室には誰もいない時（以下、夜間不在時と記す）」、「夜間、対象居室で寝ている時（以下、夜間就寝時と記す）」、「朝、起床した直後の時間帯（以下、朝起床直後と記す）」、「日中～就寝前の時間帯で、睡眠以外の目的で対象居室に在室している時（以下、起居在室時と記す）」、「日中～就寝前の時間帯で、対象居室には誰もいないが、別室で過ごしている時（以下、別室在室時と記す）」、「日中、誰も家にいない時（以下、日中外出時と記す）」の6つに区分し、時間区分ごとに回答を得た。

表 117 設問項目の一覧

質問項目	回答方法	入力値/選択肢
世帯属性	居住地域	入力 市町村
	性別	選択 男性、女性
	年齢	入力 数値
	職業	選択 勤め人、自営業者、農林水産業者、アルバイト・パート、専業主婦・主夫、小学生、中学生、高校生、その他
	世帯年収	選択 200万円未満、200～400万円未満、400～600万円未満、600～800万円未満、800～1000万円未満、1000～1200万円未満、1200～1500万円未満、1500万円以上、分からない/答えたくない
住宅建物	建築年	入力 数値
	住宅形式	選択 戸建、集合
	総階数	入力 数値
	居間/主寝室のある階	入力 数値
	住戸の位置	選択 中住戸、角住戸
住居環境に対する印象	風通し	選択 良い、やや良い、どちらとも言えない、やや悪い、悪い
	日当たり	選択 良い、やや良い、どちらとも言えない、やや悪い、悪い
周辺環境に対する印象	騒音	選択 気にならない、あまり気にならない、どちらとも言えない、やや気になる、非常に気になる
	大気汚染、砂ぼこり	選択 気にならない、あまり気にならない、どちらとも言えない、やや気になる、非常に気になる
	臭気	選択 気にならない、あまり気にならない、どちらとも言えない、やや気になる、非常に気になる
	人通り(日中・夜間)	選択 少ない、やや少ない、どちらとも言えない、やや多い、多い
	車通り(日中・夜間)	選択 少ない、やや少ない、どちらとも言えない、やや多い、多い
	治安	選択 良い、やや良い、どちらとも言えない、やや悪い、悪い
回答日から過去1週間における通風・冷房行為	夜間就寝の頻度※1	選択 週0～3回、週4～7回
	時間区分毎※2	選択 常にエアコンを使用、エアコンと窓開けのどちらかを選んで使用、エアコンは使用せず常に窓を開放、エアコンは使用せず適宜窓を開放、エアコンは使用せず常に窓を閉鎖、そのような場面はない、いずれでもない

※1 「週0～3回」である場合は「夜間不在時」、「週4～7回」である場合は「夜間就寝時」を回答

※2 時間区分は「夜間不在時」、「夜間就寝時」、「朝起床直後」、「起居在室時」、「別室在室時」、「日中外出時」の6区分

実施期間は、気象庁の天気予報を日々確認することで外気温度の異なる3期間を選定し、2013年8月9日～19日(第1回)、同年9月13日～17日(第2回)、同年10月23日～10月25日(第3回)とした。この時、第1回は盛夏、第2回は晩夏、第3回は秋を想定して、日平均気温の大よその目安をそれぞれ30℃前後、25℃前後、20℃前後とした。また、雨や台風による通風・冷房行為への影響が小さくなるように可能な範囲で配慮した。

調査終了後に設問間における回答の不整合を確認することによりデータクリーニングを行ったところ、それぞれの調査期間における有効回答数は表118のようになった。なお、アンケートの分析は統計解析ソフトSPSS Statistics 20 for Windows (IBM) を使用して行った。

表 118 解析対象世帯数

期間	回答者数	有効回答数 (回答者の男女比)
第1回	3599	1753 (51:49)
第2回	3090	1573 (51:49)
第3回	2472	1197 (52:48)

いずれかの期間の回答が有効:
2571
(51:49)
全期間の回答が有効:
487
(52:48)

4.2.3 解析対象世帯の基本属性

4.2.3.1 世帯属性

アンケート対象世帯のうち、有効な回答が得られた世帯（以下、解析対象世帯と記す）の居住地域割合は大阪府 43.4%、兵庫県 26.4%、京都府 13.1%、奈良県 7.7%、滋賀県 6.3%、和歌山県 3.0% であった。これは、近畿 2 府 4 県における 3 人～5 人世帯の世帯数^{xxxvii}の比率に近い。世帯人員数別には、3 人世帯 21.9%、4 人世帯が 60.6%、5 人世帯が 17.5% であった。

世帯属性として、世帯人員の年齢を図 270 に、職業を図 271 に、世帯年収を図 272 に示す。なお、() 内の数値は、カテゴリーごとのサンプル数を示している。世帯人員の属性は、夫と妻については全体の 6 割程度が 40 歳代であり、夫は 9 割程度が「勤め人」、妻は約半分が「専業主婦」であった。子については 4 割程度が「小学生」であった。また、世帯年収は、「400 万～600 万円」および「600 万円～800 万円」が多かった。なお、以上の世帯属性については、住宅形式（戸建住宅/集合住宅）による違いは顕著ではなかった。

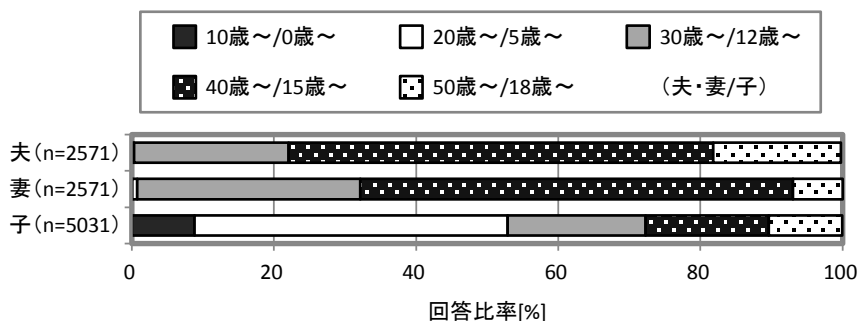


図 270 世帯人員の年齢

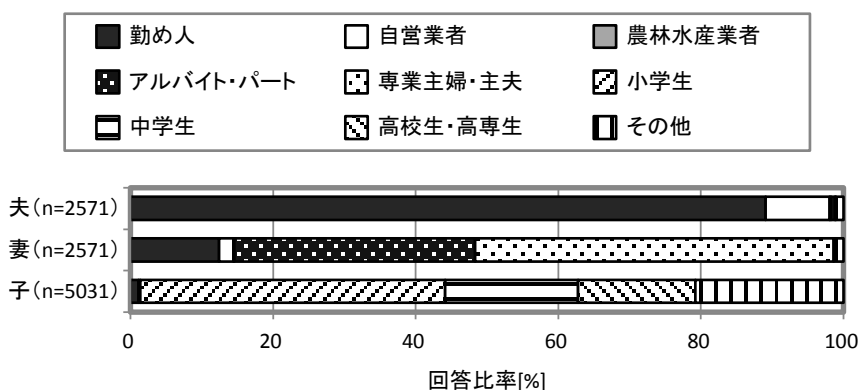


図 271 世帯人員の職業

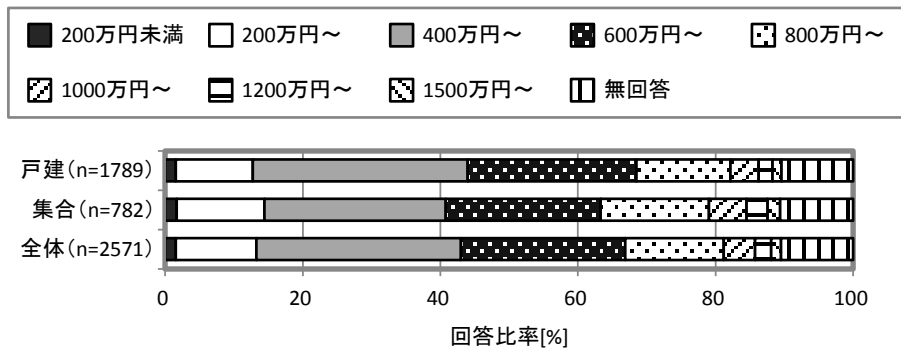


図 272 世帯年収

4.2.3.2 入居中の住宅建物

住宅形式別・住戸位置別の解析対象世帯数を表 119 に示す。ここで、住戸位置のうち階層については、1~2 階を低層階、3~5 階を中層階、6~19 階を高層階、20 階以上を超高層階と定義した。

住宅形式は、戸建住宅の方が集合住宅より多く、戸建住宅：集合住宅が大よそ 7 : 3 であった。住戸位置は、超高層階に該当する世帯が極めて少なかった。超高層階に分類された世帯は、21 階、23 階、28 階に位置し、高層階に比較的近いことから、以降の解析では超高層階に区分された世帯を高層階に含めて解析を行うこととした。

床面積は、居間については戸建住宅で「15~30 m²」、集合住宅で「15~25 m²」の回答が多く、主寝室については戸建住宅、集合住宅ともに「0~20 m²」の回答が多かった。住宅形式による床面積の差異については両居室ともに戸建住宅の方が集合住宅より大きかった。

表 119 住宅形式別・住戸位置別の解析対象世帯数

住宅形式と住戸位置			解析対象世帯数		
戸建住宅			1789		
集合住宅			782		
低層階 (1~2階)	中間階	中住戸	82	180	210
		角住戸	98		
	最上階	中住戸	10	30	
		角住戸	20		
中層階 (3~5階)	中間階	中住戸	130	243	302
		角住戸	113		
	最上階	中住戸	25	59	
		角住戸	34		
高層階 (6~19階)	中間階	中住戸	120	202	263
		角住戸	82		
	最上階	中住戸	32	61	
		角住戸	29		
超高層階 (20階~)	中間階	中住戸	2	5	7
		角住戸	3		
	最上階	中住戸	1	2	
		角住戸	1		

4.2.3.3 住居環境に関する印象

居間および主寝室の住居環境に対する印象に関する回答結果として、風通しを図 273 に、日当たりを図 274 に示す。なお、集合住宅については、住戸位置別に整理した結果を合わせて示す。

風通しについては、居間、主寝室ともに「良い」、「やや良い」が半数以上を占めたことから、大半の世帯は比較的良い通風環境にあるという印象を有していると言える。住宅形式による差異については、居間では集合住宅の方が戸建住宅よりも肯定的な回答が多かった。また、集合住宅の住戸位置による違いについては、中住戸よりも角住戸で、また階層が上がるほど肯定的な回答が占める割合が高かった。

日当たりについては、居間では「良い」、「やや良い」が半数以上を占めた。一方、主寝室では、戸建住宅については肯定的な回答が大多数であったが、集合住宅においては「どちらとも言えない」～「悪い」といった中間的・否定的な回答が半数強を占めた。また、集合住宅の中住戸/角住戸による違いについては、居間では差異はなかったが、主寝室では角住戸の方が肯定的な回答が多かった。階層による違いについては、居間においては階層が上がるほど「良い」と回答する世帯が多かったが、主寝室では差異は小さかった。

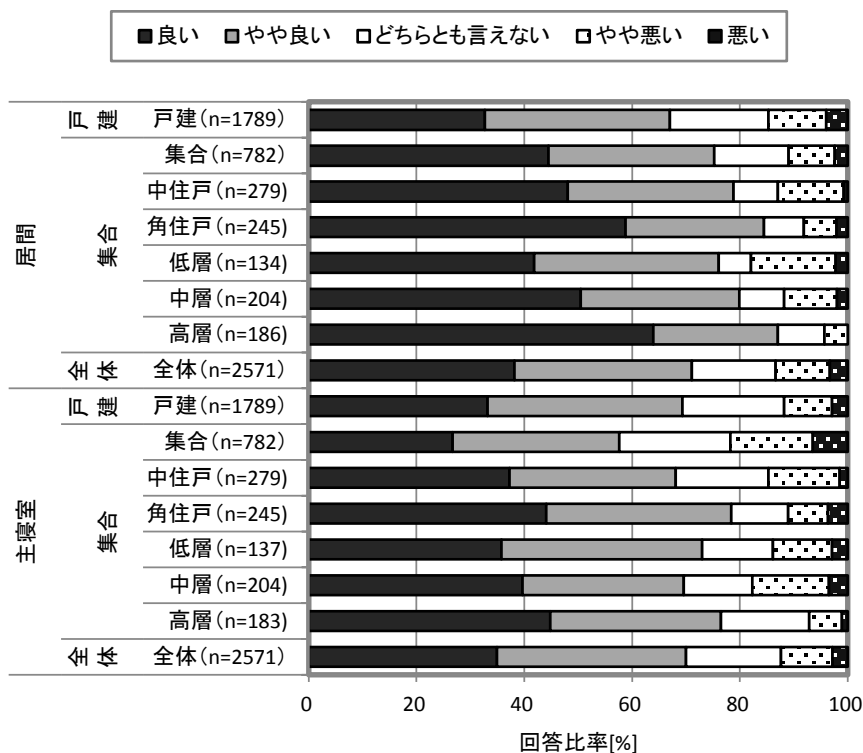


図 273 風通しに対する印象

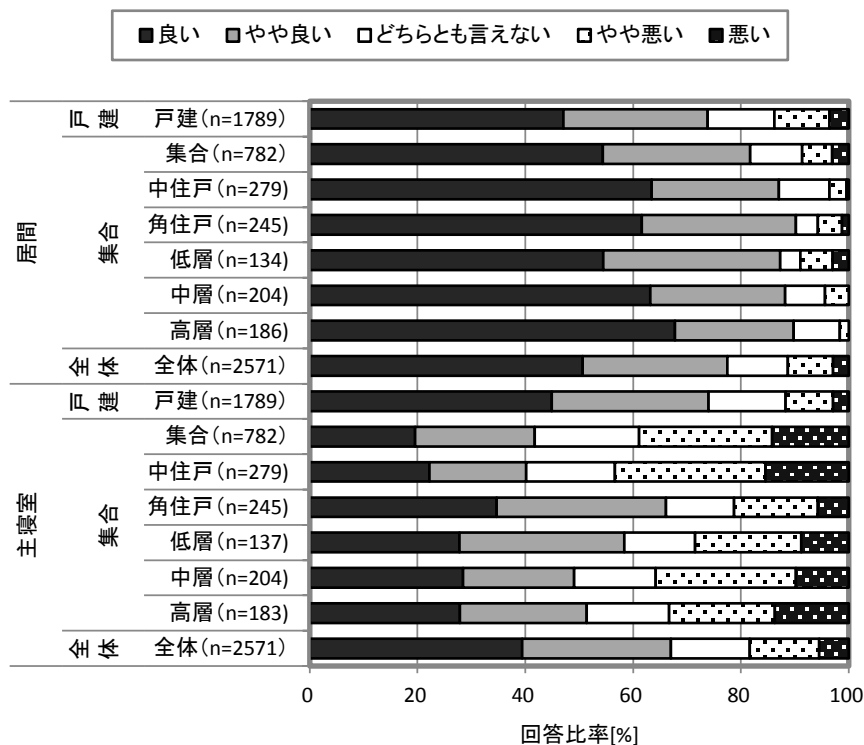


図 274 日当たりに対する印象

4.2.3.4 周辺環境に対する印象

周辺環境に対する印象に関する回答結果として、騒音・大気汚染・臭気を図 275 に、日中および夜間の人通り・車通りを図 276 に、治安を図 277 に示す。

騒音、大気汚染、臭気に対する印象については、「気にならない」、「あまり気にならない」という回答が半数以上を占めていた。人通りや車通りについては、戸建住宅の方が集合住宅よりも「少ない」、「やや少ない」という回答が多く、特に夜間にその割合が大きかった。また、治安に対しては「良い」、「やや良い」という回答が多かった。

以上のことから、解析対象世帯は、窓を開けることを阻害する心的要因は比較的少ない環境にあると考えられる。また、これらの項目については、戸建住宅の方が集合住宅よりも肯定的な回答が多いことから、戸建住宅の方が通風利用に有利な周辺環境にあると言える。

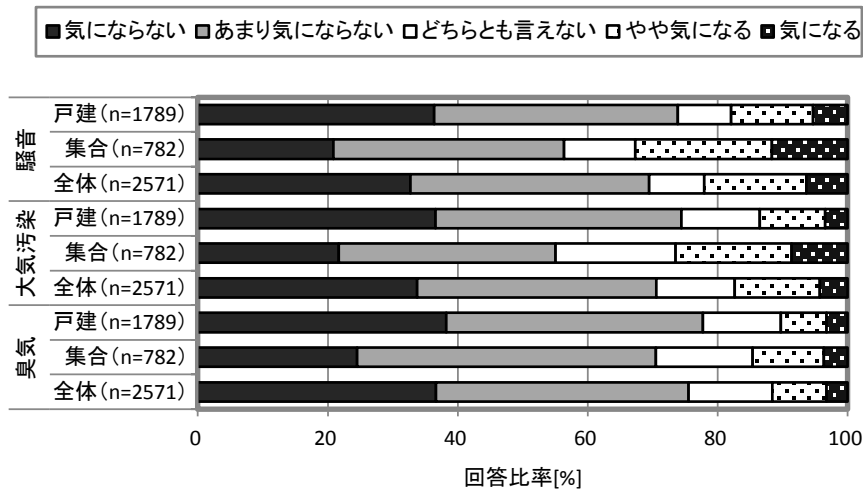


図 275 騒音・大気汚染・臭気に対する印象

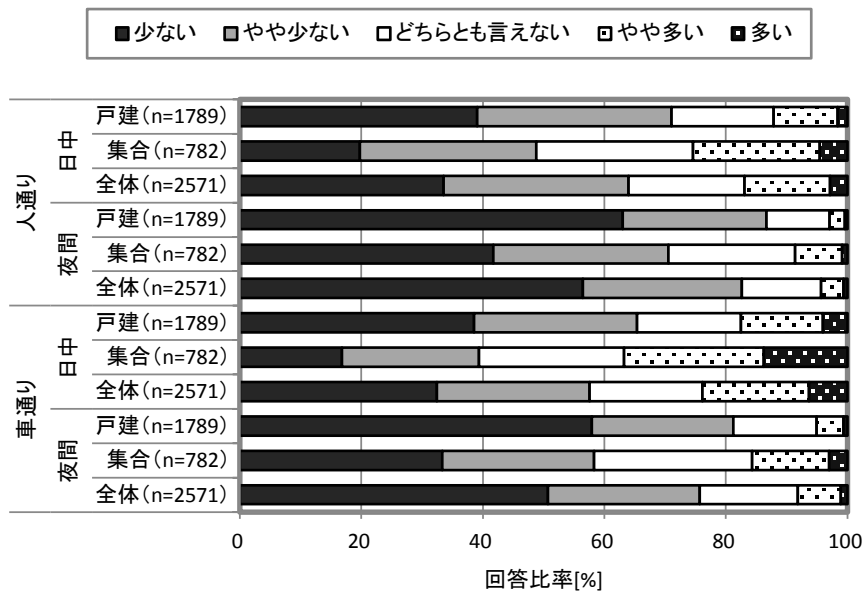


図 276 人通り・車通りに対する印象

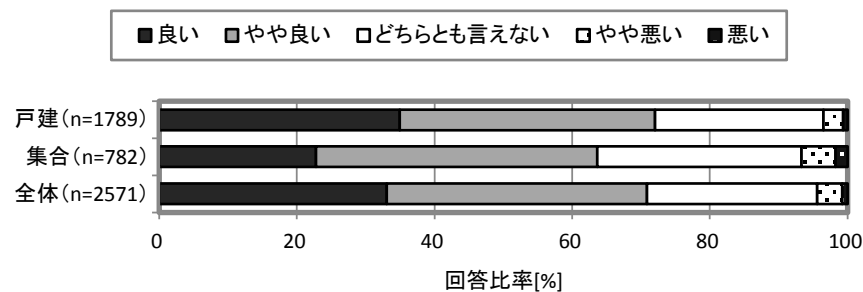


図 277 治安に対する印象

4.2.4 通風・冷房行為の選択に関するアンケートの単純集計結果

居住者が選択した通風・冷房行為を生活時間ごとに聴取した結果を調査期間別に整理して図 278 および図 279 に示す。ただし、居室の用途を鑑みて、生活時間のうち「夜間不在時」については居間のみを、「夜間就寝時」については主寝室のみを示した。なお、調査期間による相違を比較するため、全期間の回答が有効であった世帯（487 世帯）を分析対象として第 1 回から第 3 回までの解析対象世帯を統一した。また、通風・冷房行為に対する回答が「そのような場面はない」、「いずれでもない」であるものを除いた。

「朝起床直後」については、「常に窓を開放」の回答比率が他の時間区分に比べて高く、盛夏（第 1 回）であっても居間では 43.1%、主寝室では 39.2%であることから、季節を問わず、窓開けへの習慣性が強いことがうかがえる。一方で、「夜間不在時」、「夜間就寝時」および「日中外出時」においては、「常にエアコンを使用」あるいは「常に窓を閉鎖」という回答が大きな割合を占めた。これに対し、「別室在室時」は「常に窓を開放」あるいは「適宜窓を開放」を合わせると全体の半数以上となり、「常に窓を閉鎖」を大きく上回った。これは、在宅者がいる場合は在室していない室であっても窓を開放して通風利用を行う世帯が多く存在することを示している。

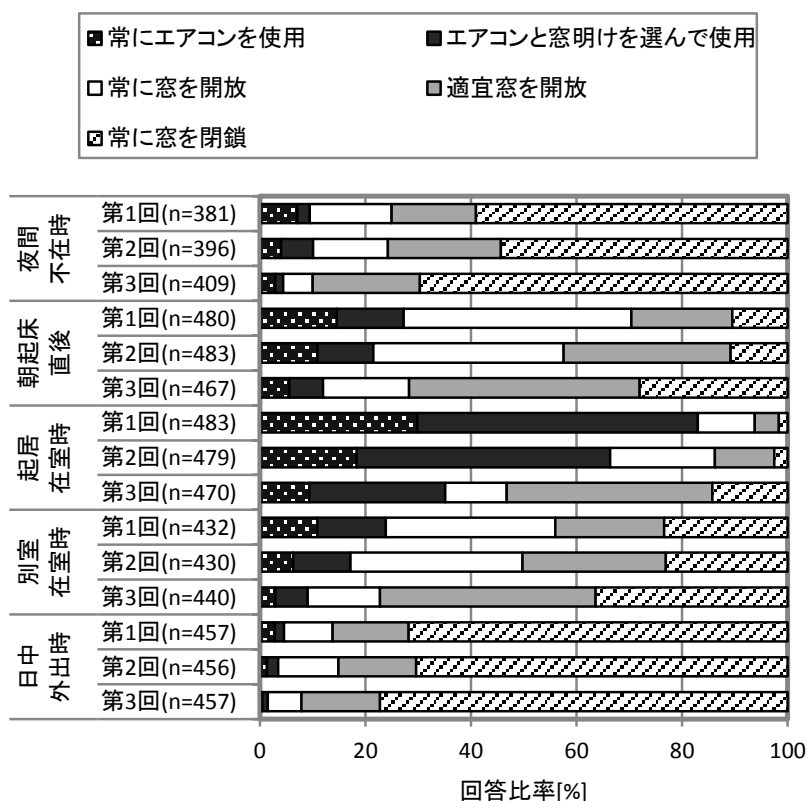


図 278 居間における通風・冷房行為の選択

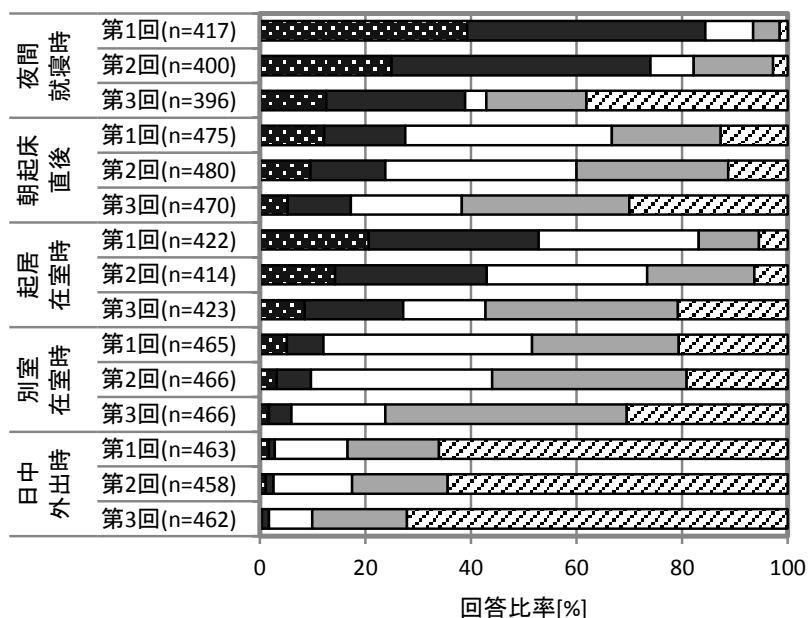
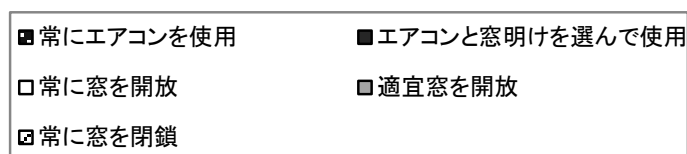


図 279 主寝室における通風・冷房行為の選択

調査期間による行為選択の相違について、同等性の検定 (χ^2 検定) および残差分析を行った結果を表 120 および表 121 に示す¹。居間、主寝室ともに「日中外出時」では 5%水準で、その他の時間区分では 1%水準で調査期間による相違に有意性が認められた。また、残差分析の結果から、盛夏(第1回)と比べて暑さが緩む晩夏(第2回)ではエアコン使用を含む回答が少なくなり、窓開けを含む回答が多くなるが、十分に涼しくなった秋(第3回)では窓開けから窓閉鎖に移行する傾向が「日中外出時」を除くすべての時間区分においてみられた。これらの結果は、「朝起床直後」といった行為選択に対する習慣性が強いことが予想される時間区分や、「別室在室時」といった居住者の熱的快適性と直接的な関係がない時間区分においても、外気温度や室内温度などの季節変化を伴う環境要素が行為選択に影響を与えていることを示唆している。

¹ 調整済み残差とは、 χ^2 検定の結果が有意であった場合にどの項目が有意性に貢献したかを判断する数値である。残差の絶対値が 2.58 より大きければ 1%水準で、1.96 より大きければ 5%水準で項目(ここでは調査期間)により有意な差があることを示す。また、残差が正(負)値の場合は他の項目に比べて有意に多い(少ない)ことを意味する。

表 120 調査期間による行為選択の相違に関する残差分析（居間）

時間区分	調査期間	調整済み残差					有意確率
		常にエアコンを使用	エアコンと窓明けを選んで使用	常に窓を開放	適宜窓を開放	常に窓を閉鎖	
夜間不在時	第1回	2.76	-1.23	2.84	-1.98	-1.01	0.000
	第2回	-0.69	3.79	1.91	1.33	-3.42	
	第3回	-2.02	-2.55	-4.68	0.62	4.38	
朝起床直後	第1回	3.66	2.50	6.44	-7.08	-4.28	0.000
	第2回	0.49	0.57	2.36	0.16	-4.04	
	第3回	-4.18	-3.09	-8.86	6.97	8.38	
起居在室時	第1回	7.21	5.87	-2.59	-9.49	-4.99	0.000
	第2回	-0.61	3.02	4.41	-4.75	-4.01	
	第3回	-6.65	-8.94	-1.83	14.33	9.06	
別室在室時	第1回	4.27	2.53	3.56	-5.04	-2.43	0.000
	第2回	-0.41	0.80	3.77	-1.35	-2.62	
	第3回	-3.85	-3.31	-7.28	6.36	5.02	
日中外出時	第1回	2.80	0.15	0.19	-0.17	-0.81	0.016
	第2回	-0.46	1.05	2.22	0.02	-1.62	
	第3回	-2.33	-1.19	-2.41	0.15	2.42	

1%水準

表 121 調査期間による行為選択の相違に関する残差分析（主寝室）

時間区分	調査期間	調整済み残差					有意確率
		常にエアコンを使用	エアコンと窓明けを選んで使用	常に窓を開放	適宜窓を開放	常に窓を閉鎖	
夜間就寝時	第1回	7.74	2.49	1.90	-5.89	-9.06	0.000
	第2回	-0.49	4.37	1.02	1.56	-7.85	
	第3回	-7.34	-6.91	-2.94	4.40	17.04	
朝起床直後	第1回	2.94	1.19	3.97	-3.84	-3.67	0.000
	第2回	0.50	0.27	2.33	1.05	-4.66	
	第3回	-3.45	-1.47	-6.32	2.79	8.36	
起居在室時	第1回	4.41	3.25	2.84	-6.82	-4.39	0.000
	第2回	-0.14	1.25	2.86	-1.44	-3.67	
	第3回	-4.27	-4.49	-5.69	8.25	8.04	
別室在室時	第1回	2.63	1.14	5.16	-4.96	-1.72	0.000
	第2回	-0.21	0.64	2.16	0.06	-2.69	
	第3回	-2.42	-1.78	-7.32	4.89	4.41	
日中外出時	第1回	1.64	-0.52	1.23	-0.35	-0.81	0.046
	第2回	0.02	0.52	2.04	0.23	-1.75	
	第3回	-1.66	-0.01	-3.26	0.12	2.56	

1%水準

4.2.5 諸要素による通風・冷房行為の選択への影響

4.2.5.1 クロス集計に基づく行為選択に影響を与える要素の抽出

前項の検討により、いずれの時間区分においても通風・冷房行為の選択に季節（調査期間）により相違があることが示された。本項では、季節以外の要素について通風・冷房行為の選択との関連性を分析した。ここでは、第1回から第3回までの調査期間のうち、回答が中間的であった第2回の結果を示す。

検討した要素は、世帯属性、住宅建物、住居環境、周辺環境に大別され、それぞれ以下の要素を検討した。すなわち、世帯属性は、「世帯年収」、「子職業」「妻職業」の3要素とした。住宅建物は、「建築年」、「住宅形式」、「階層」、「最上階」、「角住戸」の5要素とした。住居環境は居間および主寝室の「風通し」、「日当たり」に対する印象の2要素とした。周辺環境は、「騒音」、「大気汚染」、「臭気」、「人通り」、「車通り」、「治安」に対する印象の6要素とした。ここで、「子職業」とは小学生以下の子供が構成員に含まれるか否かであり、年齢が低いために行為選択の主体でないことや、親と同室で就寝することなどの影響を検討するために設けた。また、「妻職業」とは専業主婦であるか否かであり、日中不在時間の長短を代替する要素として含めた。「建築年」は、住宅建物が1992年の新基準の施行以降に建築されたか否かであり、大よその断熱性能を表す要素として設定した。

通風・冷房行為の選択に関する要素別のクロス集計に対し、同等性の検定（ χ^2 検定）を行った。ただし、「住宅形式」では住戸位置による影響を排除するため、集合住宅については低層階（1～2階）の世帯に限定した。また、「階層」、「最上階」、「角住戸」については住宅形式を集合住宅に限定した。「人通り」と「車通り」に関しては、「夜間不在時」「夜間就寝時」について夜間に対する回答を、その他の時間区分については日中に対する回答を用いた。なお、いくつかの要素については、表122に示すようにアンケートの回答により統合した新たな区分を設けることでクロス集計を行った。

表 122 アンケート回答と集計時の区分との対応

	世帯年収	アンケート回答	200万円未満	200～400万円未満	400～600万円未満	600～800万円未満	800～1000万円未満	1000～1200万円未満	1200～1500万円未満	1500万円以上		
		集計時の区分	「低所得」			「中所得」			「高所得」			
世帯属性	妻職業	アンケート回答	勤め人	自営業者	農林水産業者	アルバイト・パート	専業主婦・主夫	その他				
		集計時の区分	「専業主婦以外」				「専業主婦」	「専業主婦以外」				
世帯属性	子職業	アンケート回答	勤め人	アルバイト・パート	小学生	中学生	高校生・高専生	その他				
		集計時の区分	「中学生以上」		「小学生以下」	「中学生以上」		「小学生以下」				
住宅建物	建築年	アンケート回答	数値入力									
		集計時の区分	「1991年以前」			「1992年以降」						
	住宅形式	アンケート回答	戸建	集合								
		集計時の区分	「戸建」	「集合」								
	階層	アンケート回答	数値入力(居室の位置)									
		集計時の区分	「低層階」			「中層階」			「高層階」			
最上階	アンケート回答	数値入力(住棟の総階数)										
	集計時の区分	「中間階」			「最上階」							
角住戸	アンケート回答	中住戸	角住戸									
	集計時の区分	「中住戸」		「角住戸」								
居住環境	風通し	アンケート回答	良い	やや良い	どちらとも言えない	やや悪い	悪い					
		集計時の区分	「良い」				「悪い」					
居住環境	日当たり	アンケート回答	良い	あまり	どちらとも言えない	やや気になる	気になる					
		集計時の区分	「気にならない」		「気になる」							
周辺環境	騒音	アンケート回答	気にならない	あまり	どちらとも言えない	やや気になる	気になる					
		集計時の区分	「気にならない」		「気になる」							
	大気汚染	アンケート回答	少ない	やや少ない	どちらとも言えない	やや多い	多い					
		集計時の区分	「少ない」		「多い」							
	臭気	アンケート回答	良い	やや良い	どちらとも言えない	やや悪い	悪い					
		集計時の区分	「良い」				「悪い」					
人通り	アンケート回答	良い	やや良い	どちらとも言えない	やや悪い	悪い						
	集計時の区分	「良い」				「悪い」						

検定により得られた有意確率を表123に示す。ただし、居室の用途を鑑みて、生活時間のうち「夜間不在時」については居間のみを、「夜間就寝時」については主寝室のみを示した。いずれかの時間区分において行為選択の相違に1%水準で有意性が認められた要素は、居間では「建築年」、「住宅形式」、「階層」、「風通し」、「騒音」、「人通り」、「車通り」であった。行為選択に有意性が認められた時間区分は、「風通し」については「朝起床直後」、「起居在室時」、「別室在室時」といった日中～就寝前の間で在宅者がある時間区分、その他の要素については「夜間不在時」、「別室在室時」、「日中外出時」といった在室者がいない時間区分であった。一方、主寝室についてはいずれかの時間区分において行為選

択の相違に 1%水準で有意性が認められた要素は、「住宅形式」、「風通し」、「臭気」、「車通り」、「治安」であった。このうち「風通し」、「臭気」、「治安」については、主寝室における主な在室時間帯である「夜間就寝時」に有意性が認められた。

表 123 同等性の検定 (X² 検定) により得られた有意確率

調査 期間	時間区分	世帯属性			住宅建物					住居環境			周辺環境				
		世帯 年収	妻職業	子職業	建築年	住宅 形式	階層	最上階	角住戸	風通し	日当 たり	騒音	大気 汚染	臭気	人通り	車通り	治安
居間	夜間不在時	0.084	0.308	0.012	0.244	0.002	0.000	0.096	0.570	0.090	0.080	0.010	0.023	0.146	0.027	0.001	0.157
	朝起床直後	0.048	0.846	0.126	0.668	0.116	0.119	0.091	0.160	0.000	0.037	0.723	0.258	0.153	0.071	0.127	0.120
	起居在室時	0.914	0.592	0.052	0.369	0.106	0.337	0.427	0.427	0.000	0.022	0.153	0.105	0.323	0.165	0.433	0.075
	別室在室時	0.195	0.718	0.447	0.716	0.007	0.161	0.618	0.645	0.000	0.052	0.290	0.020	0.124	0.018	0.013	0.052
	日中外出時	0.409	0.876	0.033	0.006	0.001	0.000	0.476	0.038	0.196	0.115	0.101	0.378	0.066	0.002	0.013	0.199
主寝室	夜間就寝時	0.212	0.621	0.021	0.623	0.649	0.800	0.069	0.203	0.000	0.076	0.940	0.055	0.008	0.179	0.115	0.004
	朝起床直後	0.025	0.903	0.103	0.487	0.008	0.270	0.039	0.017	0.000	0.087	0.321	0.133	0.123	0.023	0.050	0.150
	起居在室時	0.433	0.912	0.165	0.951	0.002	0.066	0.149	0.789	0.037	0.732	0.872	0.231	0.078	0.207	0.007	0.541
	別室在室時	0.228	0.349	0.252	0.470	0.147	0.861	0.043	0.221	0.000	0.023	0.744	0.251	0.188	0.118	0.016	0.141
	日中外出時	0.257	0.953	0.077	0.111	0.142	0.057	0.527	0.056	0.275	0.055	0.271	0.194	0.720	0.157	0.091	0.074

1%水準

行為選択の相違に 1%水準で有意性が認められた要素・時間区分に関するクロス集計結果を居室別に図 280 および図 281 に示す。ただし、通風・冷房行為に対する回答が「そのような場面はない」「いずれでもない」であるものを除いた。

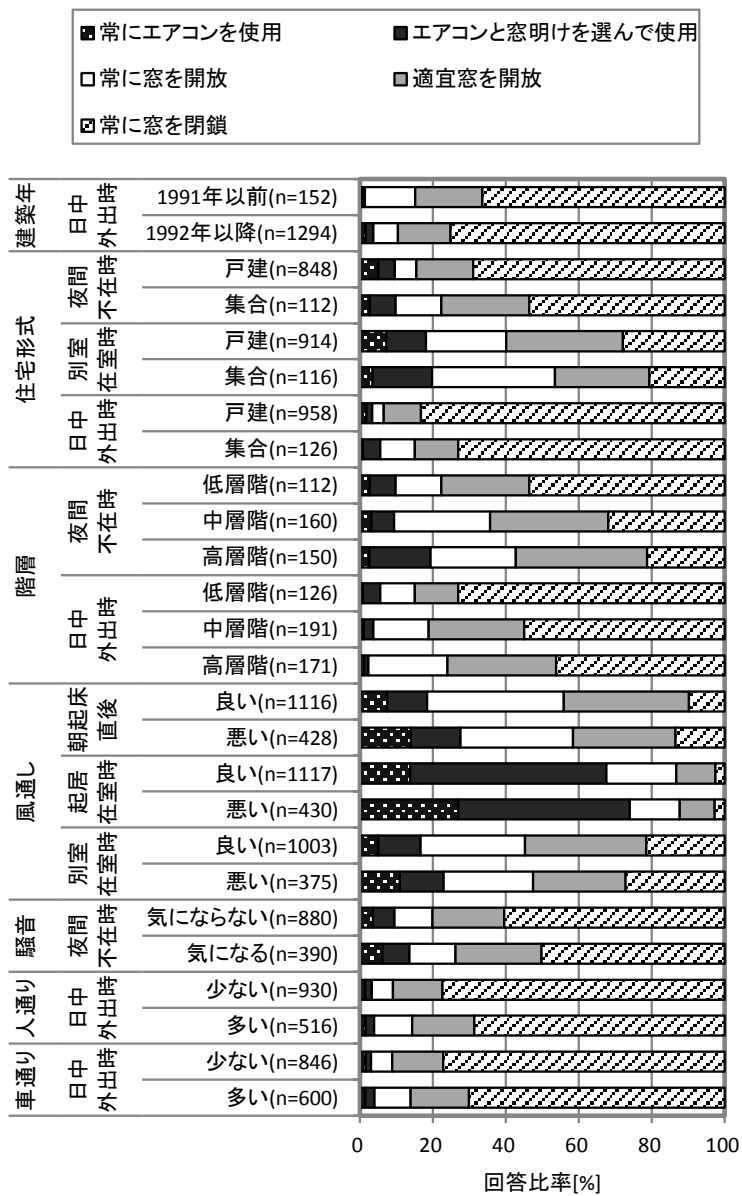


図 280 諸要素による行為選択の相違 (居間)

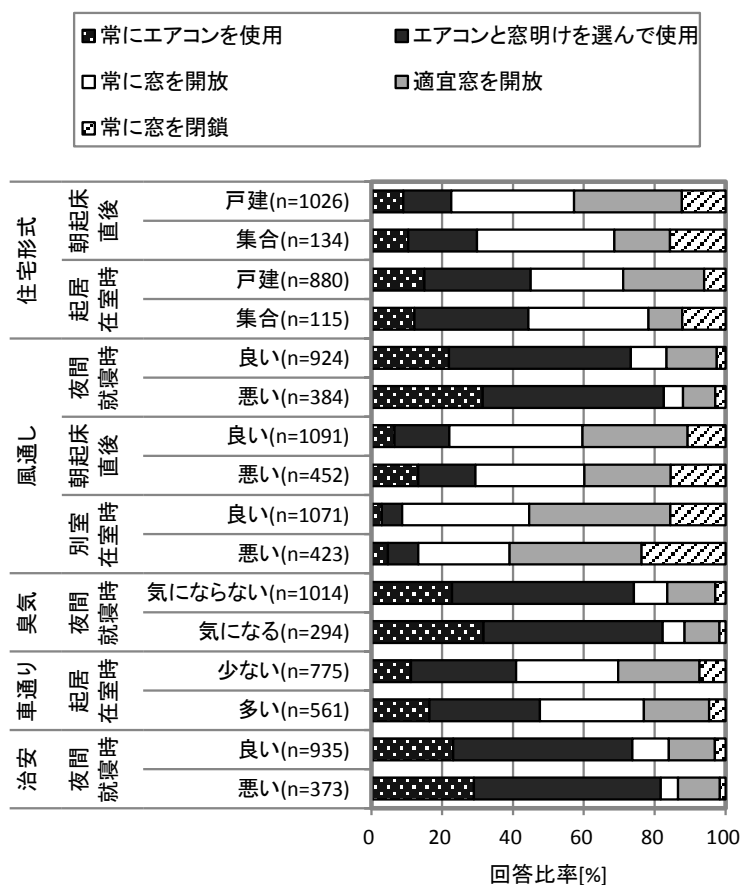


図 281 諸要素による行為選択の相違（主寝室）

「建築年」は、居間の「日中外出時」で有意性が認められた。ただし、本調査では大半が 1992 年以降に建築された住宅であり、このうち 3 分の 2 が戸建住宅であった。従って、「1992 年以降」の結果が住宅形式の違いによる影響を受けている可能性が否めず、建築年により相違があるとは断言し難い。

「住宅形式」は、居間では「戸建」において、主寝室では「集合」において他方の住宅形式に比べて窓開けを含む回答が少なく、窓閉鎖を含む回答が多かった。「住宅形式」と同じ時間区分において行為選択の相違に 5%水準で有意性が認められた要素を表 123 から探索すると、居間については「人通り」や「車通り」、主寝室については「風通し」や「車通り」であった。このうち、主寝室の風通しに対する印象は図 273 に示すように住宅形式により差がなかった。これに対し、「人通り」と「車通り」に対する印象は、図 276 に示すように戸建住宅に比べて集合住宅では「少ない」や「やや少ない」と回答した世帯が多かった。そこで、図 280 および図 281 で「人通り」と「車通り」の区分のうち「少ない」に着目すると、「日中外出時」といった在室者がいない時間区分においては窓開放が少なく窓閉鎖が多い傾向があり、一方で「起居在室時」といった在室者がある時間区分においては、エアコン使用が少なく窓閉鎖が多い傾向があった。なお、図 280 および図 281 には示していないが、居間の「夜間不在時」、「別室在室時」、主寝室の「朝起床直後」においても同様の結果を示した。同様の傾向が「住宅形式」にもみられることから、住宅形式による行為選択の相違は、人通りや車通りに対する印象に起因していると推察される。

「階層」は、居間の「夜間不在時」や「日中外出時」に対して行為選択の相違に有意性が認められた。「夜間不在時」においては、「低層階」では「常に窓を閉鎖」という回答が半数程度を占めているのに対し、「中層階」および「高層階」では「エアコンと窓開けを選んで使用」、「常に窓を開放」、「適

宜窓を開放」といった窓開放を含む回答が6割程度を占めた。同様に「日中外出時」においても、「低層階」では窓閉鎖が、上階では窓開放を含む回答が多い傾向があった。他の時間区分に比べて「夜間不在時」や「日中外出時」では防犯に対する意識が高まると考えられるが、階層が上がるほど意識が低下し、窓開放が行われやすいと考えられる。

「風通し」は、居間の「朝起床直後」、「起居在室時」、「別室在室時」、主寝室の「夜間就寝時」、「朝起床直後」、「別室在室時」において行為選択の相違に有意性が認められた。これらの時間区分においては風通しに対する印象が良い世帯は印象が悪い世帯に比べて、「常にエアコンを使用」の回答が少なく、また、「常に窓を開放」の回答が多い傾向があった。従って、居住者の風通しに対する印象が通風利用に対する積極性に影響を与えていると考えられる。

周辺環境のうち「騒音」は居間の「夜間不在時」、「臭気」は主寝室の「夜間就寝時」において1%水準で行為選択の相違に有意性が認められた。加えて「大気汚染」は居間の「夜間就寝時」において5%水準で有意性が認められた。これらの周辺環境に関する要素に対して「気になる」に区分された世帯では、居間では窓閉鎖が、主寝室ではエアコン使用が多い傾向があった。「治安」は、主寝室の「夜間就寝時」においてのみ有意性が確認された。この時間区分においては、治安に対する印象が良い世帯では「常に窓を開放」の回答が多く、「常にエアコンを使用」の回答が少なかった。以上のことから、周辺環境に関するこれらの要素は夜間における行為選択に影響をあたえると考えられる。

4.2.5.2 回帰分析に基づく行為選択に対する諸要素の重要度の検討

前項の検討では、通風・冷房行為の選択に影響を与える要素を明らかにした。本項では、それらの要素が通風・冷房行為の選択における世帯による相違の説明にどの程度寄与するかを分析した。具体的には、各時間区分における通風・冷房行為を目的変数、季節、世帯属性、住宅建物、住居環境、周辺環境といった諸要素を説明変数としてカテゴリカル回帰分析を行った。ここで、ダミー変数とカテゴリとの対応は表124のように設定した。回帰の際に説明変数として使用する要素は、前項の同等性の検定(χ^2 検定)によりいずれかの調査期間において5%水準で有意性が認められた要素とした。なお、便宜上、戸建住宅は角住戸として扱った。また、人通りと車通りに関しては、「夜間不在時」、「夜間就寝時」については夜間に対する回答を、その他の時間区分については日中に対する回答を用いた。カテゴリカル回帰分析の結果、標準化係数に5%水準で有意性が認められた要素のうち、Prattの相対重要度²が高かった3つの要素を表125に、回帰式の決定係数を表126に示す。

表 124 ダミー変数とカテゴリとの対応

ダミー変数	目的変数 通風・冷房行為	説明変数																
		季節	世帯属性			住宅建物			住居環境		周辺環境							
		調査期間	世帯年収 (単位:万円)	妻職業	子職業	建築年	住宅形式	階層	最上階	角住戸	風通し	日当たり	騒音	大気汚染	臭気	人通り	車通り	治安
1	常にエアコンを使用	第1回 (盛夏)	~200	専業主婦	小学生以下	1991年以前	戸建	低層 (1~2階)	中間階	中住戸	良い		気にならない		少ない		良い	
2	エアコンと窓開けを選んで使用	第2回 (晩夏)	200~400	専業主婦以外	中学生以上	1992年以降	集合	中層 (3~5階)	最上階	角住戸	やや良い		あまり気にならない		やや少ない		やや良い	
3	エアコンは使用せず、常に窓を開放	第3回 (秋)	400~600					高層 (6階~)			どちらとも言えない		どちらとも言えない		どちらとも言えない		どちらとも言えない	
4	エアコンは使用せず、適宜窓を開放		600~800								やや悪い		やや気になる		やや多い		やや悪い	
5	エアコンは使用せず、常に窓を閉鎖		800~1000								悪い		気になる		多い		悪い	
6			1000~1200															
7			1200~1500															
8			1500~															

² 相対重要度とは、回帰に対する説明変数の寄与率を表しており、回帰に用いた全ての説明変数のそれを積算すると1となる。具体的には、標準化係数に0次相関係数を乗じたものを回帰式の決定係数で除した値である。すなわち、世帯による行為選択の相違に関して回帰式が有する説明力に対する各要素の絶対的な重要度を表すのではなく、本研究で扱った要素の中での相対的な重要度を意味することに留意されたい。

表 125 通風冷房行為の選択に対する各要素の相対重要度

居室名称	ランク	夜間不在時		夜間就寝時		朝起床直後		起居在室時		別室在室時		日中外出時	
		要素名	重要度	要素名	重要度	要素名	重要度	要素名	重要度	要素名	重要度	要素名	重要度
居間	1	階層	■ 0.49			調査期間	☆ 0.89	調査期間	☆ 0.98	調査期間	☆ 0.54	調査期間	☆ 0.56
	2	住宅形式	■ 0.29			住宅形式	■ 0.07	風通し	▲ 0.02	住宅形式	■ 0.23	子職業	※ 0.12
	3	調査期間	☆ 0.17			治安	● 0.02			階層	■ 0.16		
主寝室	1			調査期間	☆ 0.98	調査期間	☆ 0.88	調査期間	☆ 0.92	調査期間	☆ 0.80	調査期間	☆ 0.48
	2					住宅形式	■ 0.07	住宅形式	■ 0.02	風通し	▲ 0.12		
	3					世帯年収	※ 0.01	治安	● 0.02	世帯年収	※ 0.06		

☆:「季節」、※:「世帯属性」、■:「住宅建物」、▲:「住居環境」、●:「周辺環境」

表 126 回帰式の決定係数 R²

居室名称	夜間不在時	夜間就寝時	朝起床直後	起居在室時	別室在室時	日中外出時
居間	0.16		0.15	0.25	0.10	0.01
主寝室		0.30	0.10	0.14	0.06	0.01

回帰式の決定係数は、いずれの居室・時間区分においても低い値を示した。このことは、本検討で扱った要素により表現される世帯の特徴は、世帯による通風・冷房行為の相違を説明するには十分ではないことを示している。言い換えれば、例えば「エアコンは全く使用しない」、「朝は必ず窓を開ける」といった世帯独自のルールは、本検討で扱ったような要素以外の要素により決定される部分が大いことを意味している。特に「日中外出時」は決定係数が極めて低いことから、この影響が顕著に表れている時間区分であると言える。澤島ら^{xxxvi}は『価値観は暖冷房意識を介して暖冷房の仕方に影響を与えている』と考察しているが、世帯により価値判断が大きく異なることが行為選択の相違を生じさせる大きな要因となっていると考えられる。一方で、「起居在室時」や「夜間就寝時」は、他の時間区分に比べて決定係数が比較的高かった。これらの時間区分においては、「調査期間」の重要度が0.92~0.98と極めて高いことから、居住者の熱的快適性との関連が強く、外気温度や室内温度などの環境要素により影響されやすいと考えられる。前ら^{xxx}は『住民が冷房を行う際には単に室内温度に反応するだけでなく、冷房を行う時間帯についての住戸ごとの方針が存在している』と推察しているが、本研究においても同様の結果が示された。すなわち、「起居在室時」や「夜間就寝時」では、世帯のルールを根底に外気温度や室内温度などの環境要素を加味して行為選択が行われると考えられる。「調査期間」はほとんどの時間区分において重要度が最も高い要素であったが、「夜間不在時」では「調査期間」よりも「階層」や「住宅形式」といった住宅建物に関する要素の重要度が高いという特徴がみられた。「夜間不在時」に対する回帰式の決定係数は十分に高いというわけではないが、この時間区分においては行為選択に対する世帯のルールは「階層」や「住宅形式」と関係して決定される部分があることが示唆された。

4.2.6 まとめ

本研究では、関西地域を対象としたWEBアンケートを通じて居住者が選択した通風・冷房行為を生活時間ごとに聴取し、季節、世帯属性、住宅建物、住居環境および周辺環境と通風・冷房行為との関係性を分析した。分析結果から、行為選択と関連性の強い要素を生活時間別に抽出し、要素による行為選択の相違を整理した。さらに、行為選択の相違を説明する上での要素の重要度を示した。一連の検討により得られた知見を以下に示す。

- (ア) 「朝起床直後」においては、季節を問わず、窓開けへの習慣性が強いことがうかがえた。一方で、「夜間不在時」、「夜間就寝時」および「日中外出時」においては、窓を開放することを控える傾向があった。これに対し、「別室在室時」は窓を開放して通風利用を行う世帯が多く存在した。
- (イ) 季節（盛夏、晩夏、秋の3つの調査期間）による行為選択の相違について、同等性の検定および残差分析を行った結果、居間、主寝室ともに「日中外出時」では5%水準で、その他の時間区分では1%水準で有意性が認められた。また、「朝起床直後」といった行為選択に対する習慣性が強いことが予想される時間区分や、「別室在室時」といった居住者の熱的快適性と直接的な関係がない時間区分においても、外気温度や室内温度などの季節変化を伴う環境要素が行為選択に影響を与えていることが示唆された。
- (ウ) 調査期間以外の要素でいずれかの時間区分において行為選択の相違に1%水準で有意性が認められたのは、居間では「建築年」、「住宅形式」、「階層」、「風通し」、「騒音」、「人通り」、「車通り」であった。行為選択に有意性が認められた時間区分は、「風通し」については「朝起床直後」、「起居在室時」、「別室在室時」といった日中～就寝前の中で在宅者がある時間区分、その他の要素については「夜間不在時」、「別室在室時」、「日中在室時」といった在室者がいない時間区分であった。一方、主寝室についてはいずれかの時間区分において行為選択の相違に1%水準で有意性が認められた要素は、「住宅形式」、「風通し」、「臭気」、「車通り」、「治安」であった。このうち「風通し」、「臭気」、「治安」については主寝室における主な在室時間帯である「夜間就寝時」に有意性が認められた。
- (エ) 通風・冷房行為の選択と諸要素の関係についてカテゴリカル回帰分析を行った結果、回帰式の決定係数はいずれの居室・時間区分においても低い値を示した。このことから、行為選択に対する世帯独自のルールは、本検討で扱ったような要素以外の要素により決定される部分が多いと考えられた。また、世帯による行為選択の相違を説明する上で最も重要度の高い要素は「夜間不在時」以外の時間区分においては「調査期間」であった。これに対し、「夜間不在時」では「調査期間」よりも「階層」や「住宅形式」といった住宅建物に関する要素の重要度が高かった。

本検討により、通風・冷房行為の選択に影響を与える要素や、行為選択の相違を説明する上での各要素の重要度は生活時間により異なることが示された。また、一部の生活時間区分では行為選択に対するルールに本検討で扱った要素により説明される部分があることが明らかとなった。同時に、通風・冷房行為の選択には本検討で扱った要素以外の要素が介在していることが示唆された。本検討では、物理的計測や統計資料により客観的な情報が入手可能である要素を中心に検討を行ったが、世帯により通風・冷房行為に相違が生じる要因をさらに究明するためには、主義・嗜好に関わる主観性の強い要素を含めた検討が必要である。

4.3 フィールド実測調査に基づく窓・エアコン操作の生起条件の分析

4.3.1 関連研究の動向と本調査の目的

通風・冷房行為の実態については、アンケート調査や実測調査を通じて知見の集積が行われてきた。国内では、多くの調査研究により、冷房使用や窓開放の生起率について、時間変化^{xxvi, xxvii, xxviii, xxx, xxxii, xxxiv, xxxv, xxxviii, xxxix, xl, xli}、あるいは室温や外気温等の物理量との関係性^{xxi, xxvi, xxvii, xxxix, xlii}が図示されている。さらに、一部の既往研究では、室温もしくは外気温のいずれかを変数として冷房使用の生起確率に関する予測式が示されている^{xxi, xxii, xlii}。これに対し、諸外国では実測調査データに対してロジスティック回帰分析を行うことで、冷房発停や窓開閉といった行為の生起要因を究明することが試みられている^{xv, xvii, xviii, xx, xxiii, xxiv, xliii, xliiv, xlv, xlvi}。回帰分析の説明変数には、室温、外気温、相対湿度といった物理量に加え、在室時間や入室までの不在時間といった在室行為や、窓の開閉状態などが含まれており、多様なパラメータによる検討が行われている。さらに、シチュエーション別（住宅であれば、就寝、起床、入室、退室など）に回帰分析を行っている事例^{xvi, xvii, xlvii}もみられる。ただし、諸外国（例えば、イギリス^{xv, xvii, xx, xlv}、スイス^{xviii, xxiii, xliii, xliiv}、中国^{xliv}、ドイツ^{xvi}ではオフィスの窓開放が関心の中心であったことから、住宅の窓開放^{xxiv}や冷房使用^{xlvi}に関する研究事例は未だ少ない。さらに、冷房使用と窓開放を同時に調査した事例^{xli}となれば、国内外を問わず極めて少ない。

以上のことから、通風・冷房行為のモデリングを検討する第一段階として、両行為を同時に調査した実態データを収集した上で、物理量以外の要因を含めて行為生起と要因との関係性を分析する必要があると考える。そこで、本研究では、関西地域に住まう11世帯を対象として居間と夫婦の寝室（以下、主寝室と称す）における通風・冷房行為に関する実態調査を実施した結果を各種統計的手法により分析することで、冷房発/停、窓開/閉に関する操作の生起要因を探究した³。この時、生活時間や在室行為との関係から5つの時間区分を設定し、時間区分別に分析を行うこととした。本章は、調査対象とした居室のうち、居間の分析結果について報告する。

4.3.2 調査対象および調査方法

調査概要を表127に、分析対象世帯の概略を表128に示す。本調査は、2012年7月中旬から同年10月中旬までの間で5期間に分けて実施した。分析対象世帯は、関西地域に住まう11世帯であり、一般的なファミリー世帯の構成を想定して2世代同居の3~4人世帯とした。対象居室は、居間と主寝室とした。

³ 本調査では、所有する計測機器の数量の制約により、世帯により異なる期間での調査となった。また、1調査期間あたりの日数は10日間であるが、「朝起床時_入室」や「夜間就寝前_退室」といった1日1回のみ出現する時間区分については、1世帯当たりのデータ数が少ない。以上の理由から、本調査の結果から世帯による相違について言及することは難しく、本検討では全世界帯のデータを合わせて使用し、全体的な傾向を分析することとした。

表 127 調査概要

方法	環境およびエアコン・窓のステータスデータの自動収集 ならびに記入式の在室記録
分析対象世帯	<p><u>調査対象の要件</u></p> <ul style="list-style-type: none"> ・2世代(夫婦とその子より成る)が同居する3~4人世帯 ・室内飼ペットがいない ・家族に要介護者がいない ・居間と主寝室のそれぞれにエアコンが1台以上設置されている ・エアコンと通風の両方を涼を採る手段として使う <p><u>分析対象の要件</u></p> <ul style="list-style-type: none"> ・分析が可能な期間が3期間以上である ・夏の期間(期間1・期間2・期間3)と秋の期間(期間4・期間5)からそれぞれ1期間を少なくとも含む ・入室時刻・退室時刻が1分単位で記入されている ・計測した温度データに異常な高温値が含まれない ・居間を夜間就寝のために使用していない
対象居室	居間、主寝室(夫婦の寝室)
調査期間	2012/7/17~7/26(第1回)、8/7~8/16(第2回)、 8/28~9/6(第3回)、9/18~9/27(第4回)、10/9~10/18(第5回)

表 128 分析対象世帯の概略

世帯 No	調査期間					所在	住宅 形式	建築 年	世帯主 年齢	家族構成
	1	2	3	4	5					
1	○	○		○		大阪府大阪市	戸建	2000	35	夫、妻、未就学
2	○	○	○	○	○	大阪府箕面市	集合	1975	35	夫、妻、未就学、未就学
3	○	○		○		兵庫県神戸市	戸建	2011	44	夫、妻、学生、学生
4	○	○	○	○		大阪府大阪市	戸建	1970	33	夫、妻、未就学
5	○		○		○	京都府城陽市	戸建	2006	38	夫、妻、学生、未就学
6	○		○		○	兵庫県神戸市	集合	2009	41	夫、妻、無職
7	○		○		○	大阪府大阪市	戸建	2011	38	夫、妻、未就学、未就学
8		○	○	○	○	奈良県大和高田市	戸建	1991	53	夫、妻、学生、学生
9		○		○	○	大阪府大阪市	集合	1980	46	夫、妻、学生、学生
10		○	○	○	○	奈良県大和高田市	戸建	1984	51	夫、妻、学生
11			○	○	○	京都府京都市	集合	1975	43	夫、妻、学生、学生

※学生:小~高校生

調査項目は、外気温度、室内温度、室内湿度、エアコンの稼働状況、窓の開閉状況、在室状況とした。このうち、外気温度、室内温度、室内湿度、エアコンの稼働状況、窓の開閉状況については各種計測機器により自動でデータを収集した。計測概要を表 129 に、設置例を写真 23 に示す。外気温度の計測は、ロガー付き小型温度計 (T&D TR-52i または T&D TR-52) を用いて軒下もしくはベランダで行った。室内温度および室内湿度の計測は、ロガー付き小型温湿度計 (GRAPHTEC GL10-TH、T&D TR-77Ui、T&D TR-72U のいずれか) により行った。エアコンの稼働状況は、エアコン吹き出し口にロガー付き小型温湿度計 (GRAPHTEC GL10-TK) を設置して吹出し空気温度を計測し、後の解析においてその変化により判断した)。窓の開閉状況は、磁気形接近センサー (オムロン GLS-S1) とロガー付き小型電流計 (GRAPHTEC GL10-A) により自作した記録計によりデータを収集した。計測間隔は1分を基本としたが、室内温度および室内湿度の一部は機器の記憶容量の制約から2分とした。2分値データの1分値データへの変換は、一時刻前の値により補完することで行った。なお、

機器の設置は居住者に依頼しており、外気温度、室内温度および室内湿度⁴については、直射日光やエアコン室外機・室内機およびその他家電機器からの影響ができるだけ小さい場所に設置するように依頼した⁵。一方、在室状況は、調査対象世帯に記録用紙を配布し、世帯人員別に居間または主寝室への入室/退室時刻を可能な範囲で詳細に記入するように依頼した。なお、参照する時計は時報などにより時刻を合わせるようお願いした。記入例を図 282 に示す。

表 129 計測の概要

計測項目	機器名称	センサー種別	計測箇所数	計測 間隔
外気温度	T&D TR-52i	サーミスタ	各世帯1点	1分
	T&D TR-52	サーミスタ		
室内温度 /室内湿度	T&D TR-77Ui	白金測温抵抗体 /静電容量式	各室1点	2分
	T&D TR-72	サーミスタ /高分子		2分
	GRAPHTEC GL10-TH	サーミスタ /高分子		1分
エアコンの稼働状況	GRAPHTEC GL10-TK	K型熱電対	エアコン1台につき1点	1分
窓の開閉状況	GRAPHTEC GL10-A	磁気形接近センサ	開閉箇所1つにつき1点	1分



(a) エアコン稼働状況 (b) 窓開閉状況
写真 23 計測器の設置例

⁴ 室内温度および室内湿度を計測する高さは、居住域を考慮して床上 90~120cm 程度とした。これを満たす計測場所に家具の上があるが、家具は部屋の壁際に置かれることが多く、室内環境の計測場所として必ずしも適切ではない。また、居住者の滞在場所付近に置かれるテーブル等の家具では、人体等からの発熱による影響を受ける可能性がある。以上のように、実生活環境において室内環境を適正に計測することには困難さが存在する。

⁵ 設置状況の確認は、対象世帯から提出された間取り図により行った。ここで、間取り図には対象居室の方位、エアコン・窓・ドアの位置に加え、外気温度、室内温度・湿度の計測位置が記入されている。また、設置状況を撮影した写真を提供いただいた世帯については、間取り図と合わせて確認に用いた。さらに、温度データについては、データ回収後、計測値に異常値が含まれないかを精査した。



図 282 入室時刻および退室時刻の記入例

4.3.3 分析方法

4.3.3.1 時間区分の定義

本研究では、生活時間を「朝起床時」、「起居在室時」、「夜間就寝前（主に居間）」、「夜間就寝時（主に主寝室）」の4つのシチュエーションに区分した。また、在室行為を「入室」、「滞在」、「退室」の3段階に分割した。すなわち、居間を対象とした場合の時間区分は「朝起床時_入室」、「起居在室時_入室」、「起居在室時_滞在」、「起居在室時_退室」、「夜間就寝前_退室」の計5区分である。

ここで、在室行為については「入室」、「滞在」、「退室」の時間範囲を以下の方法により設定した。入室後の経過時間、退室後の経過時間、入室までの残り時間および退室までの残り時間に対する操作の累積度数を図 283 に、入退室前後における経過時間・残り時間の定義を図 284 に示す。なお、入室前および退室後の操作は現実には発生しないが、本調査では在室状況については居住者が記録用紙に記入する方法でデータを収集したため、操作と入室および退室との間にタイムラグが存在することから、入室前および退室後についても時間範囲を設定した。ここで、操作とは、エアコンを起動する操作（以下、エアコン on 操作と記す）、エアコンを停止する操作（以下、エアコン off 操作と記す）、窓を開放する操作（以下、窓 open 操作と記す）、窓を閉鎖する操作（以下、窓 close 操作と記す）の4種類である。操作の観測度数を精査したところ、入室あるいは退室の前後 10 分以内において増加率が大きかった。そこで、本検討では入室時刻および退室時刻の前後 10 分間をそれぞれ「入室」および「退室」の時間範囲とした。

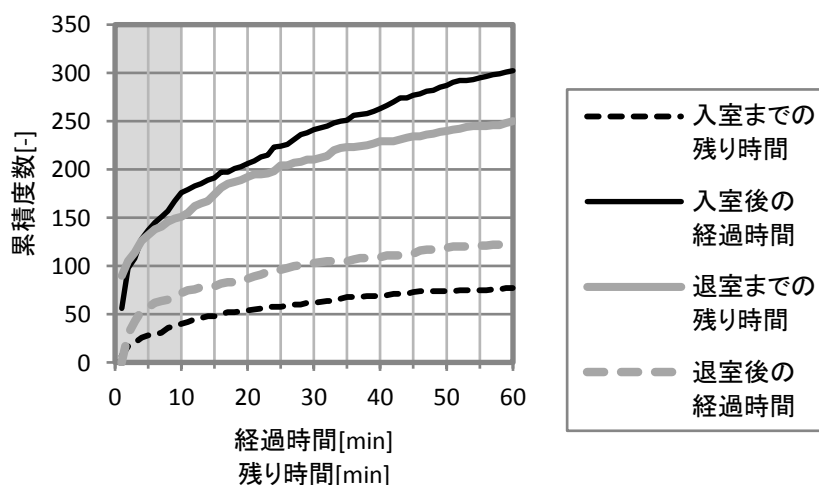


図 283 入室・退室前後において生じた操作の経過時間・残り時間に対する累積度数分布

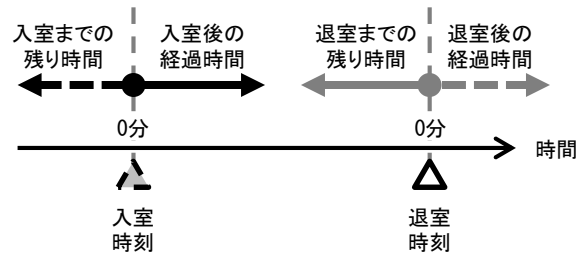


図 284 入退室前後における経過時間・残り時間の定義

4.3.3.2 統計的手法による分析方法

本検討では、統計的手法として二項ロジスティック回帰分析を適用する。二項ロジスティック回帰では、操作の生起確率 p と説明変数 x_n との間に以下のような関係を想定する。

$$\log \frac{p}{1-p} = b_0 + b_1 x_1 + \dots + b_n x_n$$

式 201

b_n : 偏回帰係数

p : 次の 10 分間で操作が生起する確率

x_n : 説明変数

ここで、「入室」および「退室」の時間範囲を入室時刻および退室時刻の前後 10 分間と定めたことに合わせて、「滞在」に対しては入室時刻を起点として 10 分毎のデータを用いた。このため、操作の生起確率は“次の 10 分間で操作が生起する確率”となる。

説明変数としては、表 130 に示すパラメータを候補とした。説明変数は、大別すると環境要素、在室行為、在室者、操作前の窓の状態である。環境要素に関するパラメータには、外気温度、室内温度、室内相対湿度の瞬時値に加え、瞬時データより算出した室内絶対湿度、室内空気比エンタルピー、前日の外気温度の平均値（以下、前日の平均気温と記す）、当日の外気温度の最低値（以下、当日の最低気温と記す）、前日の外気温度の最高値（以下、前日の最高気温と記す）を用いた。在室行為に関するパラメータは、入室から現在までの経過時間（以下、在室の経過時間と記す）、退室までの残り時間（以下、在室の残り時間と記す）、入室から退室までの時間（以下、在室の継続時間と記す）、一つ前の在室行為において退室してから現在の在室行為において入室するまでの時間（以下、直前の不在の継続時間と記す）、現在の在室行為において退室してから次の在室行為において入室するまでの時間（以下、直後の不在の継続時間と記す）とした。在室行為に関するパラメータの定義を図 285 に示す。ただし、在室行為の段階に応じて異なるパラメータを説明変数として採用した。すなわち、「入室」に対しては直前の不在の継続時間と在室の継続時間、「滞在」に対しては在室の経過時間と在室の残り時間、「退室」に対しては在室の継続時間と直後の不在の継続時間を用いた。在室者はその組み合わせに関するパラメータであり、該当する場合を 1、該当しない場合を 0 とするダミー変数とした。操作前の窓の状態は、エアコン操作が生起する直前の窓の状態であり、open の場合を 1、close の場合を 0 とするダミー変数とした。

以上のパラメータから回帰分析に用いる説明変数を選定するため、パラメータと操作の生起/不生起との関連性やパラメータ間の相関性を検証した。具体的には、数値変数（環境要素、在室行為）に対しては、操作の生起/不生起による母平均の差の検定（t 検定）を行い、5%水準で有意性が認められたパラメータを抽出した。さらに、相関分析により得られた相関係数の絶対値が 0.8 以上のパラメータは、強相関の関係にあるとみなして除外した。カテゴリー変数（在室者、操作前の窓の状態）に対し

ては、クロス集計の結果に対して独立性の検定（ χ^2 検定、もしくは Fisher の直接法）を行い⁶、5% 水準で有意性が認められたパラメータを抽出した。

表 130 分析に用いたパラメータ

大項目	小項目	備考	
環境要素	外気温度	瞬時値	
	室内温度		
	室内相対湿度		
	環境要素	室内絶対湿度	瞬時データより算出
		室内空気比エンタルピー	
		前日の平均気温	
		当日の最低気温	
		前日の最高気温	
在室行為	在室の経過時間	詳細は図4を参照	
	在室の残り時間		
	在室の継続時間		
	直前の不在の継続時間		
	直後の不在の継続時間		
在室者	夫	在室者の組み合わせに関するパラメータ。該当する場合を1、該当しない場合を0とするダミー変数。	
	妻		
	子		
	夫妻		
	父子		
	妻子		
	夫妻子		
操作前の状態	窓	エアコン操作が生起する直前の窓の状態。openの場合を1、closeの場合を0とするダミー変数。	

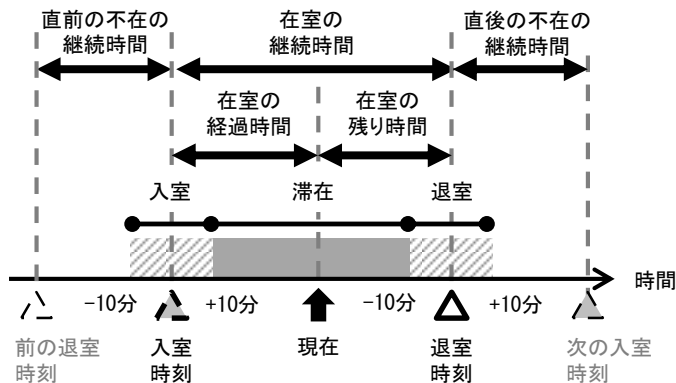


図 285 在室行為に関するパラメータの定義

以上の手順により選定した説明変数に対して回帰分析を行った。回帰分析における変数選択には、尤度比によるステップワイズ法を採用した⁷。この時、変数投入の基準は有意確率で 0.05、変数削除

⁶ 期待度数が 5 未満のセルがある場合に対して χ^2 検定を用いることは適切ではないとされている。そこで、独立性の検定は、クロス集計表のセル度数に 5 未満の値がある場合には Fisher の直接法により、それ以外の場合には χ^2 検定により行った。

⁷ 説明変数が複数あるとき、説明変数が目的変数を予測する上で重要な変数であるとは限らず、重要な変数を選択することが望ましい。本検討では、変数選択の方法としてステップワイズ法を用いた。ステップワイズ法では、まず目的変数に単独で最も寄与している変数が選択され、次に変数の追加と除去が逐次繰り返されることで変数の選択が行われる。この際、追加や除去の基準となる統計量

の基準は有意確率で 0.10 とした。ただし、窓 open 操作および窓 close 操作の解析は、窓の状態が「close」かつエアコンの状態が「on」である時のデータを除外して行った⁸。また、環境要素に係るパラメータについては、データ値が（平均値－標準偏差×2）～（平均値＋標準偏差×2）の範囲にならない場合、在室行為に係るパラメータについてはデータ値が平均値＋標準偏差を超える場合は外れ値として分析から除外した⁹。

なお、一連の分析は統計解析ソフト SPSS Statistics 20 for Windows (IBM) を使用して行った。

4.3.4 操作の生起度数と時間区分との関連性に関する分析

本節では、操作の生起と時間区分との関連性を分析した結果から、次節以降の分析を行う時間区分を選定する。操作の生起/不生起と時間区分の関連性について、クロス集計結果および独立性の検定の結果を表 131 に示す。なお、本節の時間区分に関する分析に限り、「起居在室時_滞在」の「不生起」に対しては、終始、操作が生起せず、エアコンまたは窓がある特定の状態に保持された場合を 1 カウントとした。

いずれの操作も有意水準 1%で時間区分との間に関連性が認められ、次節以降の分析を時間区分別に行う意義が確認された。操作と時間区分の関連性について全体的な傾向をみると、エアコン on 操作および窓 open 操作は「入室」と「滞在」、エアコン off 操作と窓 close 操作は「滞在」と「退室」に付随して生起した。注目すべきは、「入室」に付随して生起するエアコン on 操作と窓 open 操作であり、「朝起床時」では窓 open 操作の方が、「起居在室時」ではエアコン on 操作の方が生起度数は大きかった。このことは、「朝起床時」は窓開放に対する習慣性が存在する時間帯であることを裏付けている。

なお、生起/不生起のいずれかのデータ数が少ない時間区分が存在するため、次節以降の分析では表 131 に示すように分析対象を設定した。

(具体的には P 値) を求める方法として尤度比を用いた。なお、ステップワイズ法については回帰分析や多変量解析を扱う書籍の多くで解説されている。

⁸ エアコンの on 操作が生起する場合もしくはエアコンが on の状態にある場合は、窓の状態が一意に close と決まるため、窓の状態を判断する必要がない。また、窓の open 操作が生起する場合、エアコンが on の状態からの遷移であれば先にエアコンの off 操作を行うという判断が下された後に窓操作の判断が行われる。このことから、窓 open 操作および窓 close 操作の解析は、窓の状態が「close」かつエアコンの状態が「on」である時のデータを除外して行った。

⁹ 検討の過程において、外れ値を含む全データを使用した回帰分析を行った。しかし、外れ値による影響を受けて解釈が困難となるケースが存在した。この理由は、操作の不生起に関するデータは 10 分毎のデータであり、外れ値を含むとその値が連続的に出現することになるためである。とりわけ在室に関するパラメータ（特に継続時間に関するもの）は値そのものが大きいと、回帰分析の結果に影響を与えやすい。このため、標準偏差を用いて基準を設定し、外れ値を除外した解析を行った。

表 131 時間区分に関するクロス集計の結果

操作	生起/ 不生起 の別	上段:度数 下段:行%					合計	有意 確率
		朝起床時_入室	起居在室 時_入室	起居在室 時_滞在	起居在室 時_退室	夜間就寝 前_退室		
エアコン on	生起 off→on	15 5.9%	98 38.7%	138 54.5%	2 0.8%	0 0.0%	253 100.0%	0.000
	不生起 off→off	339 18.1%	545 29.0%	152 8.1%	571 30.4%	271 14.4%	1878 100.0%	
エアコン off	生起 on→off	2 0.8%	1 0.4%	103 41.2%	75 30.0%	69 27.6%	250 100.0%	0.000
	不生起 on→on	16 6.9%	42 18.0%	111 47.6%	43 18.5%	21 9.0%	233 100.0%	
窓open	生起 close→open	57 21.7%	35 13.3%	134 51.0%	26 9.9%	11 4.2%	263 100.0%	0.000
	不生起 close→close	177 24.8%	170 23.8%	48 6.7%	175 24.5%	143 20.1%	713 100.0%	
窓close	生起 open→close	4 2.7%	5 3.4%	80 54.1%	41 27.7%	18 12.2%	148 100.0%	0.000
	不生起 open→open	103 9.6%	335 31.3%	181 16.9%	343 32.1%	107 10.0%	1069 100.0%	

※塗りつぶしのセルは、5節以降の解析から除外した時間区分

4.3.5 操作の生起と各パラメータとの関連性に関する分析

環境要素および在室行為について、統計量および操作の生起/不生起による母平均の差の検定の結果を表 132 および表 133 に示す。また、在室者、操作前の窓の状態について、クロス集計に対する独立性の検定の結果、および残差分析の結果を表 134、表 135、表 136 に示す。

表 132 環境要素に関する統計量および母平均の差の検定の結果

操作	時間区分	統計量	外気温度 [°C]		室内温度 [°C]		室内相対湿度 [%]		室内絶対湿度 [g/kg(DA)]		室内空気比エンタルピー [kJ/kg(DA)]		前日の平均気温 [°C]		当日の最低気温 [°C]		前日の最高気温 [°C]	
			生起	不生起	生起	不生起	生起	不生起	生起	不生起	生起	不生起	生起	不生起	生起	不生起	生起	不生起
エアコン on	起居在室時入室	度数	98	510	98	510	98	510	98	510	98	510	90	454	98	510	90	454
		平均値	30.7	26.9	30.1	27.5	63.9	63.2	17.2	14.8	74.0	65.2	29.3	25.4	25.5	21.5	35.5	30.8
		標準偏差	3.42	4.30	1.41	2.83	5.33	8.51	1.67	3.42	5.22	11.17	1.77	3.88	1.72	4.56	2.53	4.22
		標準誤差	0.3	0.2	0.1	0.1	0.5	0.4	0.2	0.2	0.5	0.5	0.2	0.2	0.2	0.2	0.3	0.2
		有意確率	0.000	**	0.000	**	0.267		0.000	**	0.000	**	0.000	**	0.000	**	0.000	**
エアコン on	起居在室時滞在	度数	138	10087	138	10087	138	10087	138	10087	138	10087	122	9442	138	10087	122	9442
		平均値	29.7	26.1	30.0	27.8	66.0	64.8	17.7	15.5	75.2	67.2	28.9	25.7	25.2	21.6	34.4	31.7
		標準偏差	2.70	4.31	1.29	2.38	6.33	8.20	1.85	3.28	5.45	10.44	1.83	3.46	1.81	4.15	2.54	3.70
		標準誤差	0.2	0.0	0.1	0.0	0.5	0.1	0.2	0.0	0.5	0.1	0.2	0.0	0.2	0.0	0.2	0.0
		有意確率	0.000	**	0.000	**	0.033	*	0.000	**	0.000	**	0.000	**	0.000	**	0.000	**
エアコン off	起居在室時滞在	度数	103	7155	103	7155	103	7155	103	7155	103	7155	93	6468	103	7155	93	6468
		平均値	29.1	30.2	27.7	28.0	61.8	58.6	14.4	13.9	64.5	63.5	29.2	28.9	25.3	25.2	34.8	34.9
		標準偏差	3.16	3.41	1.79	1.30	6.58	6.20	2.03	1.71	6.55	5.16	2.12	1.93	1.95	1.80	2.81	2.72
		標準誤差	0.3	0.0	0.2	0.0	0.6	0.1	0.2	0.0	0.6	0.1	0.2	0.0	0.2	0.0	0.3	0.0
		有意確率	0.001	**	0.055		0.000	**	0.011	*	0.134		0.191		0.572		0.702	
エアコン off	起居在室時退室	度数	75	41	75	41	75	41	75	41	75	41	68	33	75	41	68	33
		平均値	30.8	30.4	28.0	27.8	59.0	59.5	14.0	13.9	63.8	63.4	29.1	29.5	25.5	25.2	35.2	35.4
		標準偏差	2.77	3.91	1.04	0.93	5.65	6.17	1.61	1.44	4.73	4.01	1.69	2.18	1.62	1.85	2.52	2.44
		標準誤差	0.3	0.6	0.1	0.1	0.7	1.0	0.2	0.2	0.5	0.6	0.2	0.4	0.2	0.3	0.3	0.4
		有意確率	0.587		0.224		0.611		0.779		0.598		0.371		0.296		0.696	
窓 open	朝起床時入室	度数	57	183	57	183	57	183	57	183	57	183	52	167	57	183	52	167
		平均値	24.8	21.9	28.6	26.0	65.5	66.4	16.2	14.3	70.0	62.5	27.2	24.9	23.6	20.5	32.4	31.1
		標準偏差	2.15	4.91	1.54	3.16	6.30	5.43	2.12	2.93	6.59	10.48	2.11	4.12	2.11	4.74	2.53	4.70
		標準誤差	0.3	0.4	0.2	0.2	0.8	0.4	0.3	0.2	0.9	0.8	0.3	0.3	0.3	0.4	0.4	0.4
		有意確率	0.000	**	0.000	**	0.278		0.000	**	0.000	**	0.000	**	0.000	**	0.000	**
窓 open	起居在室時入室	度数	35	152	35	152	35	152	35	152	35	152	30	136	35	152	30	136
		平均値	29.0	25.7	28.6	26.5	64.8	65.1	16.2	14.4	69.9	63.3	26.6	24.4	22.2	20.0	32.8	30.8
		標準偏差	4.07	4.58	2.20	3.14	7.89	7.12	3.12	3.34	9.86	11.39	2.93	4.48	3.49	5.30	3.45	4.62
		標準誤差	0.7	0.4	0.4	0.3	1.3	0.6	0.5	0.3	1.7	0.9	0.5	0.4	0.6	0.4	0.6	0.4
		有意確率	0.000	**	0.000	**	0.821		0.005	**	0.002	**	0.001	**	0.003	**	0.008	**
窓 open	起居在室時滞在	度数	134	2992	134	2992	134	2992	134	2992	134	2992	127	2923	134	2992	127	2923
		平均値	25.5	24.0	27.6	26.6	65.1	65.8	15.3	14.6	66.6	64.0	26.1	24.8	22.0	20.3	31.9	31.1
		標準偏差	4.66	4.68	2.08	2.60	7.41	6.90	2.84	3.27	9.02	10.76	3.68	3.99	4.20	4.85	4.11	3.87
		標準誤差	0.4	0.1	0.2	0.0	0.6	0.1	0.2	0.1	0.8	0.2	0.3	0.1	0.4	0.1	0.4	0.1
		有意確率	0.000	**	0.000	**	0.278		0.014	*	0.001	**	0.000	**	0.000	**	0.011	*
窓 close	起居在室時滞在	度数	80	7081	80	7081	80	7081	80	7081	80	7081	72	6506	80	7081	72	6506
		平均値	24.7	26.9	27.4	28.3	64.0	64.4	15.0	15.8	65.7	68.6	24.8	26.1	20.8	22.2	30.6	32.0
		標準偏差	4.44	3.83	2.48	2.07	9.74	8.66	3.81	3.23	11.94	9.99	3.08	3.09	4.07	3.68	3.45	3.59
		標準誤差	0.5	0.0	0.3	0.0	1.1	0.1	0.4	0.0	1.3	0.1	0.4	0.0	0.5	0.0	0.4	0.0
		有意確率	0.000	**	0.002	**	0.746		0.067		0.030	*	0.000	**	0.001	**	0.001	**
窓 close	起居在室時退室	度数	41	322	41	322	41	322	41	322	41	322	37	288	41	322	37	288
		平均値	28.7	27.0	28.9	27.8	65.8	63.0	16.7	15.0	71.6	66.2	27.1	25.9	23.0	22.4	32.4	30.9
		標準偏差	2.95	3.99	1.81	2.56	7.33	8.69	2.97	3.40	9.21	10.86	2.74	3.56	3.28	4.11	3.33	4.05
		標準誤差	0.5	0.2	0.3	0.1	1.1	0.5	0.5	0.2	1.4	0.6	0.5	0.2	0.5	0.2	0.5	0.2
		有意確率	0.002	**	0.001	**	0.051		0.002	**	0.001	**	0.024	*	0.274		0.034	*

表 133 在室行為に関する統計量および母平均の差の検定の結果

操作	時間区分	統計量	在室の経過時間[min]		在室の残り時間[min]		在室の継続時間[min]		直前の不在の継続時間[min]		直後の不在の継続時間[min]	
			生起	不生起	生起	不生起	生起	不生起	生起	不生起	生起	不生起
エアコンon	起居在室時_入室	度数					98	510	98	510		
		平均値					280	227	173	191		
		標準偏差					194.6	211.8	227.6	221.2		
		標準誤差					20	9	23	10		
		有意確率					0.016 *		0.473			
	起居在室時_滞在	度数	138	10087	138	10087						
		平均値	226	236	314	268						
		標準偏差	246.3	213.9	228.0	244.9						
		標準誤差	21	2	19	2						
		有意確率	0.631		0.029 *							
エアコンoff	起居在室時_滞在	度数	103	7155	103	7155						
		平均値	379	304	135	252						
		標準偏差	274.9	235.0	177.0	201.9						
		標準誤差	27	3	17	2						
		有意確率	0.007 **		0.000 **							
	起居在室時_退室	度数					75	41			74	40
		平均値					309	261			153	95
		標準偏差					216.5	170.2			131.3	107.7
		標準誤差					25	27			15	17
		有意確率					0.226				0.018 *	
窓open	朝起床時_入室	度数					56	182	53	168		
		平均値					417	251	422	407		
		標準偏差					336.5	318.7	92.2	142.5		
		標準誤差					45	24	13	11		
		有意確率					0.001 **		0.457			
	起居在室時_入室	度数					35	152	35	152		
		平均値					273	260	257	182		
		標準偏差					203.2	231.5	161.8	200.5		
		標準誤差					34	19	27	16		
		有意確率					0.761		0.043 *			
起居在室時_滞在	度数	134	2992	134	2992							
	平均値	127	233	283	202							
	標準偏差	153.6	228.2	254.2	205.9							
	標準誤差	13	4	22	4							
	有意確率	0.000 **		0.000 **								
窓close	起居在室時_滞在	度数	80	7081	80	7081						
		平均値	284	238	166	296						
		標準偏差	228.6	207.6	191.6	254.7						
		標準誤差	26	2	21	3						
		有意確率	0.048 *		0.000 **							
	起居在室時_退室	度数					41	322			41	322
		平均値					220	153			263	161
		標準偏差					92.4	155.0			147.6	167.5
		標準誤差					14	9			23	9
		有意確率					0.000 **				0.000 **	

表 134 在室者に関するクロス集計および独立性の検定の結果

操作	時間区分	生起/不生起の別	度数	相対度数[%]							上段:有意確率 下段:Cramer 連関係数
				夫	妻	子	夫妻	夫子	妻子	夫妻子	
エアコンon	起居在室時_入室	生起	98	2.0	21.4	13.3	0.0	2.0	42.9	18.4	0.000
		不生起	510	13.7	38.2	24.5	1.0	0.6	19.6	2.4	0.37 **
	起居在室時_滞在	生起	138	3.6	11.6	8.7	7.2	2.9	39.1	26.8	0.005
		不生起	10087	8.8	18.8	13.0	6.7	3.7	30.2	18.7	0.04 **
エアコンoff	起居在室時_滞在	生起	103	21.4	8.7	8.7	9.7	7.8	19.4	24.3	0.000
		不生起	7155	5.5	10.6	5.3	6.4	3.4	38.9	29.8	0.09 **
	起居在室時_退室	生起	75	5.3	17.3	6.7	1.3	0.0	57.3	12.0	0.141
		不生起	41	2.4	29.3	12.2	7.3	0.0	43.9	4.9	0.27
窓open	朝起床時_入室	生起	57	14.0	49.1	12.3	5.3	3.5	7.0	8.8	0.002
		不生起	181	28.2	21.0	24.9	4.4	5.5	9.9	6.1	0.29 **
	起居在室時_入室	生起	35	2.9	62.9	14.3	0.0	0.0	20.0	0.0	0.051
		不生起	152	9.2	37.5	33.6	2.0	0.7	13.2	3.9	0.26
	起居在室時_滞在	生起	134	6.0	18.7	9.7	11.2	3.7	30.6	20.1	0.001
不生起		2992	16.4	17.0	14.7	8.0	7.6	19.6	16.6	0.09 **	
窓close	起居在室時_滞在	生起	80	11.3	21.3	7.5	5.0	5.0	40.0	10.0	0.032
		不生起	7081	5.6	19.5	12.4	6.1	2.1	34.7	19.6	0.04 *
	起居在室時_退室	生起	41	17.1	46.3	2.4	2.4	0.0	22.0	9.8	0.010
		不生起	322	17.7	27.3	14.9	0.6	0.0	35.7	3.7	0.20 **

表 135 在室者に関する残差分析の結果¹⁰

操作	時間区分	調整済み残差[-]						
		夫	妻	子	夫妻	夫子	妻子	夫妻子
エアコンon	起居在室時_入室	-3.28	-3.18	-2.43	-0.98	1.46	4.98	6.70
	起居在室時_滞在	-2.14	-2.16	-1.51	0.24	-0.51	2.27	2.42
エアコンoff	起居在室時_滞在	6.84	-0.63	1.53	1.34	2.42	-4.03	-1.22
	起居在室時_退室							
窓open	朝起床時_入室	-2.16	4.14	-2.00	0.26	-0.61	-0.67	0.71
	起居在室時_入室							
	起居在室時_滞在	-3.22	0.49	-1.61	1.31	-1.68	3.10	1.06
窓close	起居在室時_滞在	2.18	0.38	-1.32	-0.42	1.80	0.99	-2.15
	起居在室時_退室	-0.10	2.51	-2.20	1.21	0.00	-1.75	1.77

多い(5%水準)	少ない(5%水準)
----------	-----------

¹⁰ 調整済み残差とは、独立性の検定結果が有意であった場合にどの項目が有意性に貢献したかを判断する数値である。残差の絶対値が2.58より大きければ1%水準で、1.96より大きければ5%水準で項目により有意な差があることを示す。また、残差が正(負)値の場合は他の項目に比べて有意に多い(少ない)ことを意味する。

表 136 エアコン on 操作前の窓の状態に関するクロス集計および独立性の検定の結果

操作	時間区分	生起/不生起の別	度数	操作前の窓の状態[%]		有意確率(χ^2 検定)	Φ 係数
				close	open		
エアコン on	起居在室時_入室	生起	98	69.4	30.6	0.000	-0.26 **
		不生起	510	34.7	65.3		
	起居在室時_滞在	生起	138	39.1	60.9	0.021	-0.02 *
		不生起	10087	29.7	70.3		

環境要素については、エアコン off 操作の「起居在室時_退室」ではいずれのパラメータとの関連性も認められなかった。エアコン off 操作の「起居在室時_退室」を除くと、環境要素のうち外気温度や室内温度等の温度環境に関連する要素はいずれの操作・時間区分においても生起/不生起による相違が認められた。一方で、室内相対湿度に相違が認められたのは、エアコン on 操作の「起居在室時_滞在」およびエアコン off 操作の「起居在室時_滞在」であった。

在室行為については、エアコン on 操作の「起居在室時_入室」および窓 open 操作の「朝起床時_入室」では在室の継続時間、エアコン on 操作の「起居在室時_滞在」では在室の残り時間、エアコン off 操作の「起居在室時_退室」では直後の不在の継続時間に生起/不生起による相違が認められた。これらの操作・時間区分では、各パラメータの平均値は「生起」の方が「不生起」に比べて大きかった。従って、居住者は操作以降の在室の継続時間や不在の継続時間を予想して、操作を行うか否かを判断しているものと考えられる。窓 open 操作の「起居在室時_入室」では、直前の不在の継続時間に生起/不生起による相違が認められた。その平均値は「生起」の方が「不生起」に比べて大きいことから、入室前の不在時間が長い場合には窓を開けて空気の入れ換えが行われていると推測できる。エアコン off 操作、窓 open 操作および窓 close 操作の「起居在室時_滞在」では在室の経過時間および在室の残り時間の両パラメータに生起/不生起による相違が認められた。経過時間は「生起」の方が「不生起」に比べて、窓 open 操作では短く、エアコン off 操作および窓 close 操作では長かった。一方で、残り時間は「生起」の方が「不生起」に比べて、窓 open 操作では長く、エアコン off 操作および窓 close 操作では短かった。これらを考え合わせると、窓 open 操作は「滞在」の比較的早い段階で、エアコン off 操作および窓 close 操作は比較的遅い段階で生起すると言える。窓 close 操作の「起居在室時_退室」では在室の継続時間および直後の不在の継続時間の両パラメータに生起/不生起による相違が認められた。いずれのパラメータも「生起」の方が「不生起」に比べて長いことから、短時間での頻繁な窓操作は起こりにくいと考えられる。

在室者については、エアコン off 操作の「起居在室時_退室」、窓 open 操作の「起居在室時_入室」以外の操作・時間区分において生起/不生起による相違が認められた。残差分析の結果から、エアコン on 操作では、夫、妻、子がそれぞれ単独で在室する場合には操作が起こりにくく、妻子を含む場合には操作が起こりやすい¹¹ことが示唆された。エアコン off 操作の「起居在室時_滞在」では、在室者の組み合わせが夫のみもしくは夫と子の場合に操作が生起することが多い傾向にあった。また、窓 open 操作の「朝起床時_入室」、窓 close 操作の「起居在室時_退室」では妻が単独で在室する場合に操作が起こることが多かった。

エアコン on 操作前の窓の状態については、生起/不生起による相違が認められた。クロス集計の結果から、「起居在室時_滞在」で生起するエアコン on 操作の 60.9%は窓が開いた状態から生起した。これは、暑さへの対応として、まずは窓開放が優先的に行われ、暑さを凌ぐことが難しくなればエアコン使用に遷るという順番が存在することを裏付ける結果である。

¹¹ 低年齢の子を含む世帯では、子が大人の居住者（特に妻）と行動を共にする傾向にあった。残差分析において「妻子」および「夫妻」で有意性が認められたが、こうした対象世帯の属性が影響した可能性は否定できない。

4.3.6 回帰分析による操作の生起に対する各変数の影響度の検討

環境要素および在室行為に関するパラメータについては、さらに相関分析を行い、前項の母平均の差の検定の結果と合わせて、回帰分析に用いる説明変数を選定した。なお、相関分析において外気温度と室内温度が強相関の関係にある場合には、室内温度を優先的に選択した。相関分析の詳細は割愛するが、説明変数の選定結果をその過程を含めて表 137 に整理する。

表 137 回帰分析に用いる説明変数の選定結果

操作	時間区分	環境要素							在室行為					在室者	操作前の窓の状態	
		外気温度	室内温度	室内相対湿度	室内絶対湿度	室内空気比エンタルピー	前日の平均気温	当日の最低気温	前日の最高気温	現在室の経過時間	現在室の残り時間	現在室の継続時間	直前の不在の継続時間			直後の不在の継続時間
エアコンon	起居在室時_入室	○	●		○	○	○	○	●			●			●	●
	起居在室時_滞在	●	●	●	○	○	○	○	●		●				●	●
エアコンoff	起居在室時_滞在	●		●	●					●	●				●	
	起居在室時_退室													●		
窓open	朝起床時_入室	○	●		○	○	○	○	●			●			●	
	起居在室時_入室	○	●		○	○	○	○	●			●			●	
	起居在室時_滞在	●	●		○	○	○	○	●	●	●				●	
窓close	起居在室時_滞在	●	●			○	○	●	●	●	●				●	
	起居在室時_退室	●	●		○	○	○		●					●	●	





-  分析対象から除外した変数
-  検定により操作との関連性が示唆されなかった変数
-  検定により操作との関連性が示唆された変数のうち、相関分析により除外された変数
-  検定により操作との関連性が示唆された変数のうち、回帰分析に用いた変数

表 139 回帰式による生起確率の予測における各パラメータの設定値

パラメータ	標準値
外気温度	28 °C
前日の最高気温	32 °C
在室の経過時間	180 min
在室の残り時間	
在室の継続時間	360 min
直前の不在の継続時間	300 min
直後の不在の継続時間	
在室者	妻子
操作前の窓の状態	close

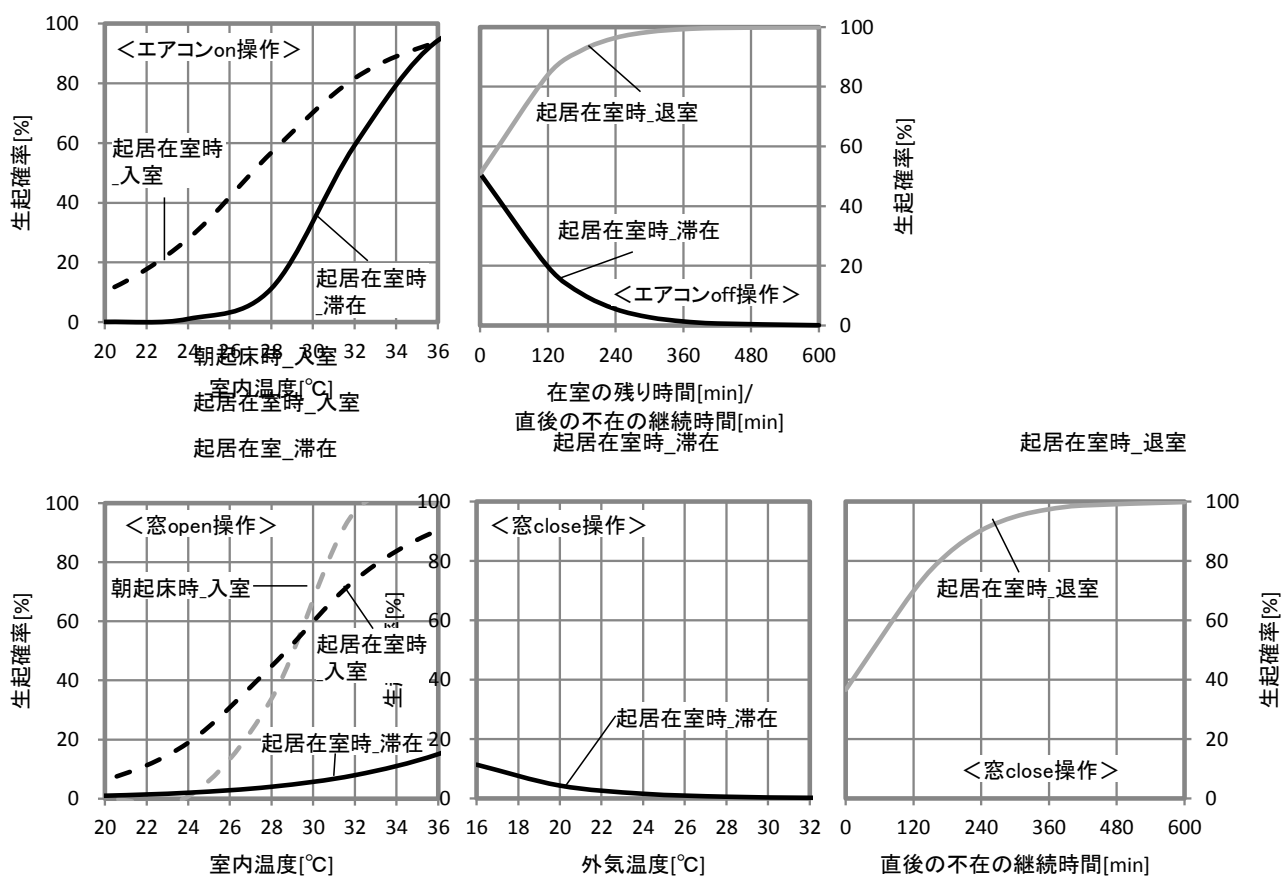


図 286 回帰式による生起確率の予測の一例

エアコン on 操作について、「起居在室時_入室」では在室者の標準化偏回帰係数が大きく、操作生起に対する影響度は環境要素よりも高かった。在室者の偏回帰係数は、妻子および夫妻で正の値であることから、妻子を含む在室者の組み合わせでは入室に伴うエアコン on 操作の生起確率が高まると言える。また、前日の最高気温に対する標準化偏回帰係数は比較的大きく、また偏回帰係数は正の値であった。このことは、前日の最高気温が高いほど、入室に伴う on 操作の生起確率が高いことを示している。従って、入室時には前日の最高気温を参考に入室以降の温熱環境の状況を予想してエアコン使用を行うと考えられる。一方で、「起居在室時_滞在」では室内温度の標準化偏回帰係数が他の変数に比べて大きく、室内温度は滞在中のエアコン on 操作の生起に対して重要な要素と言える。また、在室者については、妻および夫の偏回帰係数は負の値、夫妻のそれは正の値であったことから、大人の居住者は在室者が少ない場合にエアコン使用を控える傾向があることや、在室人数が多いにはエアコン使用が行われやすい傾向があることがうかがえる。「起居在室時_入室」と「起居在室時_滞在」の間で共通する傾向として、エアコン操作前の窓の状態に対する標準化偏回帰係数が比較的大きく、その偏回帰係数は正の値であった。このことは、操作前の窓の状態が「open」である場合は「close」である場合に比べてエアコン on 操作の生起確率が低くなることを示している。

エアコン off 操作の「起居在室時_滞在」では在室の残り時間が説明変数として選択され、偏回帰係数は負の値であった。「起居在室時_退室」では、直後の不在の継続時間が説明変数として選択され、偏回帰係数は正の値であった。このことは、在室の残り時間が短いほど、また直後の不在の継続時間が長いほどエアコン off 操作が行われやすいことを意味している。

窓 open 操作は、いずれの時間区分においても室内温度に対する標準化偏回帰係数が大きく、とりわけ「朝起床時_入室」の値が大きかった。「朝起床時_入室」では、前日の最高気温に対する標準化偏回帰係数が大きいことから、入室に伴うエアコン on 操作と同様に入室以降の温熱環境の状況を予想して通風利用を行うと考えられる。一方で、「起居在室時_滞在」では在室者の夫に対する標準化偏回帰係数が大きく、負の値であった。このことから、夫は窓が閉まっているならその状態を保持する傾向にあると考えられる。

窓 close 操作については、「起居在室時_滞在」では外気温度に対する標準化偏回帰係数が最も大きかった。ただし、本調査で観察された温度範囲では操作生起に対する外気温度の影響は小さかった。一方、「起居在室時_退室」では在室の継続時間および直後の不在の継続時間が説明変数として選択され、操作に対する影響力は同程度であった。また、偏回帰係数はともに正の値であることから、在室や不在の継続時間がある程度短い場合には窓閉鎖は生起しにくく、窓開放が継続される傾向にあると言える。

4.3.7 まとめ

本研究では、関西地域に住まう 11 世帯を対象として、通風・冷房行為に関する実態調査のうち居間の調査結果について各種統計的手法により分析した。分析では 5 つの時間区分を設定し、エアコン発/停、窓開/閉に関する行為の生起要因を時間区分ごとに探究した。一連の検討により得られた知見を以下に示す。

- (ア) 生活時間と在室行為との組み合わせにより、5 つの時間区分（「朝起床時_入室」、「起居在室時_入室」、「起居在室時_滞在」、「起居在室時_退室」、「夜間就寝前_退室」）を設定し、操作の生起/不生起との関連性について独立性の検定を行った。その結果、いずれの操作も有意水準 1% で時間区分との間に関係性が認められ、回帰分析を時間区分別に行う意義が確認された。また、エアコン on 操作および窓 open 操作は「入室」と「滞在」、エアコン off 操作と窓 close 操作は「滞在」と「退室」に付随して生起した。
- (イ) 環境要素、在室行為、在室者、操作前の窓の状態に関するパラメータに対し、操作の生起/不生起との関連性に関して検定や相関分析を行い、ロジスティック回帰分析に使用するパラメータを選定した。
- (ウ) 回帰分析の結果、エアコン on 操作について、「起居在室時_入室」では在室者の標準化偏回帰係数が大きく、操作生起に対する影響度は環境要素よりも高かった。「起居在室時_滞在」では室内温度の標準化偏回帰係数が最も大きかった。「起居在室時_入室」と「起居在室時_滞在」の間で共通する傾向として、操作前の窓の状態に対する標準化偏回帰係数が比較的大きく、操作前の窓の状態が「close」の方が「open」よりも生起確率は高かった。
- (エ) エアコン off 操作の「起居在室時_滞在」では在室の残り時間が説明変数として選択され、偏回帰係数は負の値であった。また、「起居在室時_退室」では、直後の不在の継続時間が説明変数として選択され、偏回帰係数は正の値であった。
- (オ) 窓 open 操作は、いずれの時間区分においても室内温度に対する標準化偏回帰係数が大きく、とりわけ「朝起床時_入室」の値が大きかった。室内温度以外のパラメータでは、「朝起床時_入室」では前日の最高気温、「起居在室時_滞在」では在室者の夫に対する標準化偏回帰係数が大きかった。
- (カ) 窓 close 操作については、「起居在室時_滞在」では外気温度に対する標準化偏回帰係数が最も大きかった。一方、「起居在室時_退室」では在室の継続時間および直後の不在の継続時間が説明変数として選択され、操作に対する影響力は同程度であった。

本報では、多様なパラメータに対して統計的手法により分析を行うことで、環境要素以外のパラメータがエアコン操作および窓操作の生起に寄与していることを示した。なお、本分析は 11 世帯を対象として実施したものであり、結果の一般性を担保しているとは言い難い。しかしながら、統計的な手法を適用することにより、多様なパラメータから通風・冷房行為の要因を特定した事例はこれまでにはなく、本検討は通風・冷房行為のモデリングに対して有用な知見を示せたと考える。

4.4 通風・冷房行為に関するモデルの考案

4.2 節の WEB アンケート調査結果から、「起居在室時」や「夜間就寝時」のみならず、「朝起床直後」といった行為選択に対する習慣性が強いことが予想される時間区分や、「別室在室時」といった居住者の熱的快適性と直接的な関係がない時間区分においても、外気温度や室内温度などの季節変化を伴う環境要素が行為選択に影響を与えていることが示唆された。これに対し、「別室在室時」は「常に窓を開放」あるいは「適宜窓を開放」を合わせると全体の半数以上となり、「常に窓を閉鎖」を大きく上回った。以上のことから、通風・冷房行為のパターンとして、「夜間不在時」および「日中外出時」においては「常に窓を閉鎖」、それ以外の生活時間帯では何かしらのトリガーによりエアコンと窓の操作が生起すると仮定できる。

4.3 節の調査結果に基づき、各操作の生起に対するトリガーとその閾値を検討した。なお、モデルの検討においては単純化のため、標準化回帰係数の絶対値が大きな値をとることが多かった外気温度および室内温度に要素を限定して検討した。検討結果を整理して表 140 に示す。エアコンの on 操作は入室時と滞在時に生起することが多く、室内温度の影響が大きいことが示唆された。エアコンの on 操作が生起した時の室内温度の平均値は、窓閉鎖状態から生起する場合には 29.8℃、窓開放状態から生起する場合には 30.1℃であった。また、両者の間の差異は有意水準 5%で有意であり、その差の 95%信頼区間は 0.02~0.61℃であった。エアコンの off 操作は、環境要素との関連が見られず、退室に伴い生起するものと考えられた。窓の open 操作は入室時と滞在時に生起することが多く、室内温度の影響が大きいことが示唆された。窓の close 操作は、滞在時と退室時に生起し、滞在時に生起する場合は外気温度の影響が大きいことが示唆された。窓の open 操作が生起した時の室温条件は窓開放時（通風時）の室内温度から検討することとし、その平均値は 28.1℃であった。滞在時に窓の close 操作が生起した時の外気温度の平均値は、24.2℃であった。

表 140 各操作の生起に対するトリガーとその閾値

操作	エアコン		窓		
	on	off	open	close	
操作が生起する 在室行為	滞在・入室		滞在 入室	滞在	退室
操作が生起する トリガー	室内 温度		室内 温度	外気 温度	退室
操作が生起する 直前の窓の状態	閉	開			
サンプル数	118	104	7326	76	
平均値	29.8	30.1	28.1	24.2	
標準偏差	1.0	1.2	1.9	3.9	
平均値の標準誤差	0.09	0.12	0.13	0.44	
有意確率	0.038 *				
差の 95% 信頼区間	0.02 ~ 0.61				

以上を踏まえて、図 287 に示す状態遷移図に従うモデルの構築を行った。ただし、本結果は 11 世帯の調査データより検討したものであるため、今後も継続してデータの蓄積を行い、精査する必要がある。

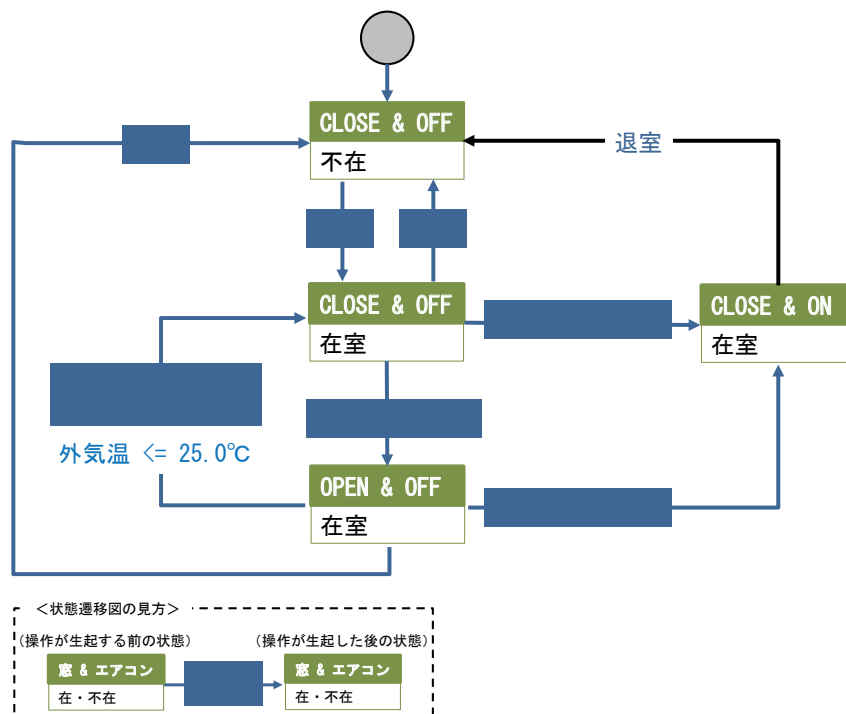


図 287 状態遷移図

4.5 まとめ

本章では、関西地域を対象とした WEB アンケートを通じて居住者が選択した通風・冷房行為を生活時間ごとに聴取し、季節、建物条件、住居環境および周辺環境と通風・冷房行為との関係性を分析することで、諸要素の行為選択への影響について生活時間による相違を把握した。検討の結果、行為選択に対する世帯独自のルールは、本検討で扱った客観的な要素以外の要素により決定される部分が大きいと考えられた。本検討では、物理的計測や統計資料により客観的な情報が入手可能である要素を中心に検討を行ったが、世帯により通風・冷房行為に相違が生じる要因をさらに究明するためには、主義・嗜好に関わる主観性の強い要素を含めた検討が必要である。また、世帯による行為選択の相違を説明する上で最も重要度の高い要素は「夜間不在時」以外の時間区分においては「調査期間」であった。これに対し、「夜間不在時」では「調査期間」よりも「階層」や「住宅形式」といった住宅建物に関する要素の重要度が高かった。

また、関西地域に住まう 11 世帯を対象として、通風・冷房行為に関する実態調査のうち居間の調査結果について各種統計的手法により分析した。分析では 5 つの時間区分を設定し、エアコン発/停、窓開/閉に関する行為の生起要因を時間区分ごとに探究した。検討の結果、エアコン on 操作に対しては、「起居在室時_入室」では在室者の組合せが、「起居在室時_滞在」では操作が生起する直前の窓の状態が強く影響した。エアコン off 操作に対しては、「起居在室時_滞在」では在室の残り時間が、「起居在室時_退室」では直後の不在の継続時間が説明変数として選択された。窓 open 操作の「朝起床時_入室」、「起居在室時_入室」、「起居在室時_滞在」では室内温度の影響度が大きかった。窓 close 操作に対しては、「起居在室時_滞在」では外気温度が、「起居在室時_滞在」では在室の継続時間および直後の不在の継続時間が説明変数として選択された。

以上の調査結果に基づいて、エアコンの on/off 操作および窓の open/close 操作を生起させるトリガーとその閾値を設定し、通風・冷房行為に関するモデルを考案した。

5 潜熱負荷を適切に考慮した熱負荷計算法の開発と基準原案の作成

ここまで示した建物特性・居住者の行動、空調設備の挙動を反映して建物の顕熱・潜熱負荷およびエネルギー消費量を評価するプログラムの作成と計算を行った。

5.1 プログラムの概要

プログラム言語はC#（開発環境はMicrosoft Visual Studio Community 2015）とした。

プログラム開発にあたっての基本的な考え方としては、本事業で調査検討したモデルを実装することは当然として、今後このプログラムが省エネ基準の検討において何らかの形で継続的に使用される可能性を考えた場合、計算各要素のモデル化の変更（精緻化あるいは簡略化）、新たに要請される計算要素の追加等に柔軟に対応できる形としておくことが有用である。

したがって、本プログラムでは計算要素の一つ一つをクラス化するとともに、C#におけるインターフェース（関数等の入出力の大枠を定めたもの。関数の実体はユーザーが自由に作成することができる）機能を用いて各クラス・各メソッド等の入出力関係を整理したうえで実装することとした。

熱・水分移動現象の基本的なモデル化は、

- ・ 時間進行は前進差分法による
- ・ 壁体等の温度計算は有限体積法による

とした。

プログラム内では、壁体・窓等の躯体や開口部、室（空間）、人体、空調設備（エアコン）等のように、時々刻々の未知の状態量を持つ要素を「ブロック」と称し、クラス名も***Blockのように命名している。同様に、「対流熱伝達」「換気熱・水分移動」などの熱・水分移動そのものもブロックとして扱っている。

逆にブロックではないと解釈しているものは、気象データ、スケジュール、内部発熱要素などである。

各ブロックは、建物において熱・水分的に影響を受ける他のブロックの状態量等を取得し、ブロック内の状態量を更新する。この処理を全ブロックについて行うことで計算上の時間が単位時間だけ進行することになる。

各ブロックは、原則として少なくとも以下のメソッド（処理）を行う。

- ・初期化（メソッド名 **Init**）
- ・時間進行ごとの初期化（メソッド名 **IterationInit**）
- ・時間進行（メソッド名 **Run**）
- ・時間進行ごとの後処理（メソッド名 **Commit**）

主なブロックにおける上記の各メソッドの処理を簡単に示すと下図のようになる。

IterationInit メソッドを各ブロックについて行い（下図、横方向にブロックを巡回）、次に図の下方向に進んで **Run** メソッドを各ブロックについて行い、最後に **Commit** メソッドを各ブロックに行うことで時間が一つ進行する。

	熱伝達 ブロック	壁体ブロック	窓ブロック	エアコン ブロック	人 ブロック	室 ブロック
IterationInit メソッド	無し	無し	透過日射量を 求める	処理熱量・消 費電力量を 求める	エアコン ON/OFF、窓 開け判定を行 う	内部発熱量の 集計、エアコン ON/OFF判定 を行う
Run メソッド	各種熱伝達量 を求める	温湿度を求め る	温度を求める	無し	無し	温湿度を求め る
Commit メソッド	無し	温湿度を更新 する	温度を更新す る	無し	無し	温湿度を更新 する

図 288 主なブロックにおける処理と時間進行

なお、壁体等の計算に時間前進差分を用いているため、壁体等ごとに反復計算が発散しないための最大時間刻みが存在する。中空層等を有する壁体や、もともとの熱容量が小さいガラス等はこの最大時間刻みが数秒程度となる。

全体の時間進行をこの数秒という刻みに合わせると、各クラスの呼び出し等に要する実行時間が膨大になり実用的でないため、全体の同期をとる時間刻みを1分とし、ブロックによって最大時間刻みが1分未満になる場合は各ブロックの **Run** メソッド内でも反復計算を行うようにした。

したがって、時間が1分進行するまでの間、各壁体の表面および内部温度が更新されても、外部（外気温・室温、他の壁体等の表面温度等）は更新されずに計算を行うことになる。室などに関しても同様で、例えばエアコンの **ON/OFF** 判定はこの1分の間は最初に決まった状態が（仮に室温が下がってエアコンが不要な温度になったとしても）維持されることになる。

5.2 各クラスの概要

プログラムのソースコードは付録に記載している。ここでは各クラスの主なメソッドとその処理内容の概要を示す。

5.2.1 Program クラス

(1) Main メソッド

プログラム実行時に最初に呼び出される。

計算期間・計算時間間隔等の設定、入力データの読み込みと各ブロックの生成、出力内容の設定などを行っている。

なお、入力データは Excel ファイルに記載する方式を取っている。



No.	一方の居室名 (居室A)	もう一方の居室名 (居室B) 空白の場合は 外部環境	外部環境の 場合の 方位角	外部環境の 場合の 傾斜角	面積[m2]	床フラグ Aにとって床 であれば1	壁仕様名	壁仕様 向き	A側αc [W/m2K]	A側αr [W/m2K]	A側日射 吸収率	A側長波 放射率	B側αc [W/m2K]	B側αr [W/m2K]	B側日射 吸収率	B側長波 放射率
1																
2	1 WA	FLS	0	180	16.562	1	0FLOOR	1	2	4.7	0.8	0.9	2	4.7	0.8	0.9
3	2 WA		90	90	8.736	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
4	3 WA		180	90	6.33	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
5	4 WA		180	90	2.184	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
6	5 LD	FLS	0	180	21.5306	1	0FLOOR	1	2	4.7	0.8	0.9	2	4.7	0.8	0.9
7	6 LD	WA	0	0	5.676	0	0WALL	1	4.4	4.7	0.8	0.9	4.4	4.7	0.8	0.9
8	7 LD		270	90	6.591	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
9	8 LD		180	90	7.266	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
10	9 K	LD	0	0	5.46	0	0WALL	1	4.4	4.7	0.8	0.9	4.4	4.7	0.8	0.9
11	10 K		270	90	7.483	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
12	11 K		180	90	3.49675	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
13	12 BATH		90	90	3.555	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
14	13 BATH		180	90	4.095	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
15	14 HALL		90	90	2.842	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9
16	15 HALL		180	90	3.549	0	0WALL	2	4.4	4.7	0.8	0.9	20.3	4.7	0.8	0.9

図 289 入力ファイル

5.2.2 Simulator クラス

計算時間間隔、暖房期・冷房期・中間期などの定義を保持している。

(1) Run メソッド

各ブロックの IterationInit メソッド、Run メソッド、Commit メソッドを順に呼び出している。

5.2.3 SimulatorContext クラス

計算中の日付・時刻、曜日、暖房期・冷房期・中間期の区分等を管理している。

5.2.4 IBlock インターフェイス

ブロックが持つべきプロパティ、メソッドを定義している。

5.2.5 IConnector インターフェイス

ブロック間の情報伝達に関するブロックが持つべきプロパティ、メソッドを定義している。

5.2.6 BaseBlock クラス

各ブロックに共通する要素を抽出した基底クラスである。

(1) CalcDt メソッド

各ブロックの熱水分移動計算で発散しないための最大時間刻みを求めている。

5.2.7 ISpaceBlock インターフェイス

スペース (室および外部環境) に関するブロックが持つべきプロパティ、メソッドを定義している。

5.2.8 ILayerBlock インターフェイス

表面における熱伝達等に関するブロックが持つべきプロパティ、メソッドを定義している。

5.2.9 RoomBlock クラス

室のブロックであり、温湿度等の物理量のほか、建物情報 (室が面する壁体・開口部等のインスタンス)、エアコン、人などの参照情報を保持する。

(1) Init メソッド

各変数の初期化、表面積等の集計等を行う。

(2) IterationInit メソッド

内部発熱要素の発熱量の集計、在室者または自身の温度によるエアコン ON/OFF 判定等を行う。

(3) Run メソッド

接続する壁体・開口部等からの熱水分伝達量、内部発熱量、エアコンによる処理熱量等を集計し、自身の時間進行後の温湿度を求める。

(4) Commit メソッド

(3) で求めた温湿度を確定させ、MRT (平均放射温度) を求める。

5.2.10 WallBlock クラス

壁体のブロックであり、分割した節点ごとの温湿度等の物理量のほか、表面の接続先スペース、壁体仕様などの参照情報を保持する。

(1) WallBlock メソッド (コンストラクタ)

節点を生成し、節点ごとの熱湿気容量、節点間の熱水分抵抗等を算出する。

(2) Run メソッド

壁体内の熱水分移動を解き時間進行後の温湿度を求める。

(3) Commit メソッド

(2) で求めた温湿度を確定させる。

5.2.11 WindowBlock クラス

窓のブロックであり、両側表面節点の温度等の物理量のほか、表面の接続先スペース、窓 (ガラス) 仕様などの参照情報を保持する。

(1) WindowBlock メソッド (コンストラクタ)

節点を生成し、節点ごとの熱湿気容量、節点間の熱抵抗等を算出する。

(2) IterationInit メソッド

ガラスを透過して室に到達する日射量等を計算する。

(3) Run メソッド

壁体内の熱水分移動を解き時間進行後の温湿度を求める。

(4) Commit メソッド

(3) で求めた温湿度を確定させる。

(5) AngleFactor メソッド

日射入射角を引数として入射角特性を返す

5.2.12 DoorBlock クラス

ドアのブロックであり、両側表面節点の温度等の物理量のほか、表面の接続先スペースなどの参照情報を保持する。

(1) DoorBlock メソッド (コンストラクタ)

節点を生成し、節点ごとの熱湿気容量、節点間の熱抵抗等を算出する。

(2) Run メソッド

壁体内の熱水分移動を解き時間進行後の温湿度を求める。

(3) Commit メソッド

(2) で求めた温湿度を確定させる。

5.2.13 WallFaceConnector クラス

スペース（室または外気）と壁体・窓・ドア等との接続関係を保持する。後述する対流熱伝達または総合熱伝達率ブロックと1対1に対応する。

5.2.14 OutSideBlock クラス

外界条件を保持する。

(1) Init メソッド

気象データを読み込む。

(2) IterationInit メソッド

当該時刻の気象要素値（補間値）を取得する。

(3) GetSAT メソッド

表面特性を受け取り SAT 温度を返す。

5.2.15 FixedConditionRoomBlock クラス

温湿度をユーザーが指定する室（床下等を想定）の設定を保持する。

5.2.16 PersonBlock クラス

人間を表す。一人につき PersonBlock が一つではなく、ある室に在室する複数人をまとめて一つのインスタンスとして生成する。

(1) ACONOFF メソッド

室温・外気温、自身の過去の在室状況等からエアコンの ON/OFF 判定を行う。

5.2.17 AirConditionerBlock クラス

エアコンを表す。

(1) Run メソッド

ON の場合に、処理熱量（顕熱・潜熱）および消費電力量を求める。

5.2.18 ConvectionHeatTransferBlock クラス

対流熱伝達量を求める。

5.2.19 RadiationHeatTransferBlock クラス

放射熱伝達量を求める。

5.2.20 TotalHeatTransferBlock クラス

外気側の総合熱伝達量を求める。

5.2.21 VentilationHeatTransferBlock クラス

換気による熱・水分移動量を求める。

5.2.22 ConvectionMoistureTransferBlock クラス

表面における水分移動量を求める。

5.2.23 Material クラス

壁体等の材料熱特性を保持する。

5.2.24 HeatGenerator クラス

室内内部発熱要素の情報を保持する。

5.2.25 SMASHWeatherLoader クラス

SMASH形式の気象データを読み込むほか、補間値の計算等を行う。

5.2.26 Util クラス

湿り水蒸気に関する各種関数を保持する。

5.3 計算結果

作成したプログラムを用いて、住宅の省エネ基準用モデル建物の 4～8 地域における冷房一次エネルギー消費量を求めた結果を示す。

建物側の条件は、

- ・ H11 断熱仕様
- ・ 居室間欠冷房
- ・ 居室窓はレースカーテン

とし、現行の冷房基準値が求められている条件と同じ条件とする。

結果を表 141 に示す。

表 141 冷房一次エネルギー（基準値）計算結果

	4地域	5地域	6地域	7地域	8地域
冷房一次エネルギー基準値 (現行判定プログラムによる)	1,467	1,578	4,331	4,778	8,173
室温のみによるON/OFF判定結果	1,996	1,839	4,321	4,661	6,505
人体によるON/OFF判定結果	2,356	1,954	5,587	6,203	9,128

単位[MJ/年]

ここで、「室温のみによる ON/OFF 判定結果」は、現行省エネルギー基準の設定温度（27℃、夜間は 28℃）を室温が上回る場合に冷房を運転したものの、「人体による ON/OFF 判定結果」は第 4 章に示したロジックを反映したものである。

なお、今回開発したプログラムは、住宅の省エネルギー基準の計算に使用されたプログラム（AE-Sim/Heat）の結果にできるだけ近い値が得られるよう配慮したが、いくつかの点では完全に同じ処理とすることはできない。その主なものを以下に示す。

- ・ 床下温度・小屋裏温度

本プログラムは床下・小屋裏ともに外気温同等とした。Simheat の床下は地盤温度（一般に夏期は外気温より低い）の影響を受けるため、本プログラムは安全側。SimHeat の小屋裏は日射・夜間放射を受けて気温が上昇する（換気量によりある程度相殺されるが）ため、本プログラムは日中に関しては危険側。

- ・ 外部日除け

本プログラムは日除けを無視したため夏期は安全側。

- ・ 通気層

本プログラムでは空気層として扱う。Simheat は外気との換気が行われるとしている。本プログラムの方が安全側と考えられる。

室温およびエアコンの挙動を見るため、6 地域における代表的な 2 日間の室温湿度変動およびエアコン消費電力量の変動を示す。

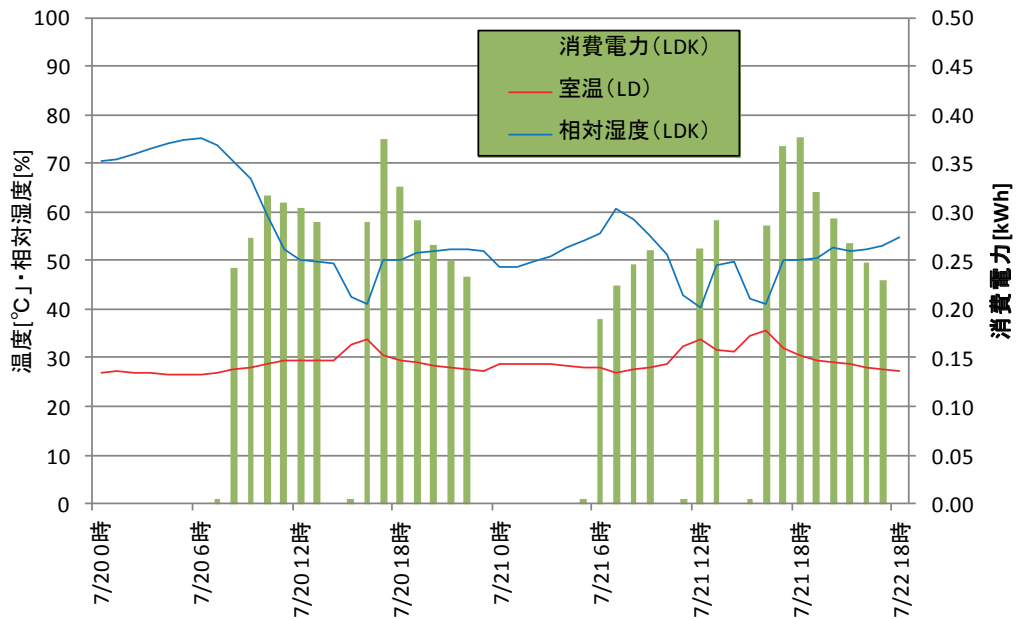


図 290 室温湿度・消費電力量変動（6 地域、LD、室温のみによる ON/OFF 判定）

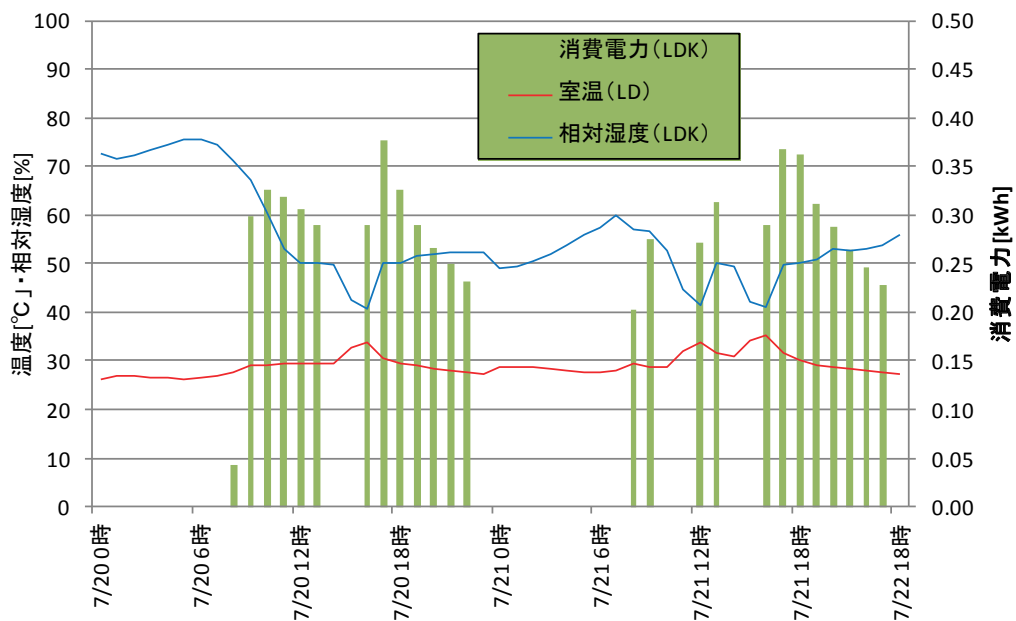


図 291 室温湿度・消費電力量変動（6 地域、LD、人体による ON/OFF 判定）

LD の結果を見ると、この 2 日間の室温は冷房中でも設定温度である 27°C まで低下することはほとんどなく、やや高い温度で推移している。一方相対湿度は設定湿度である 60% を下回り、おおむね 50% 程度が維持されていることがわかる。

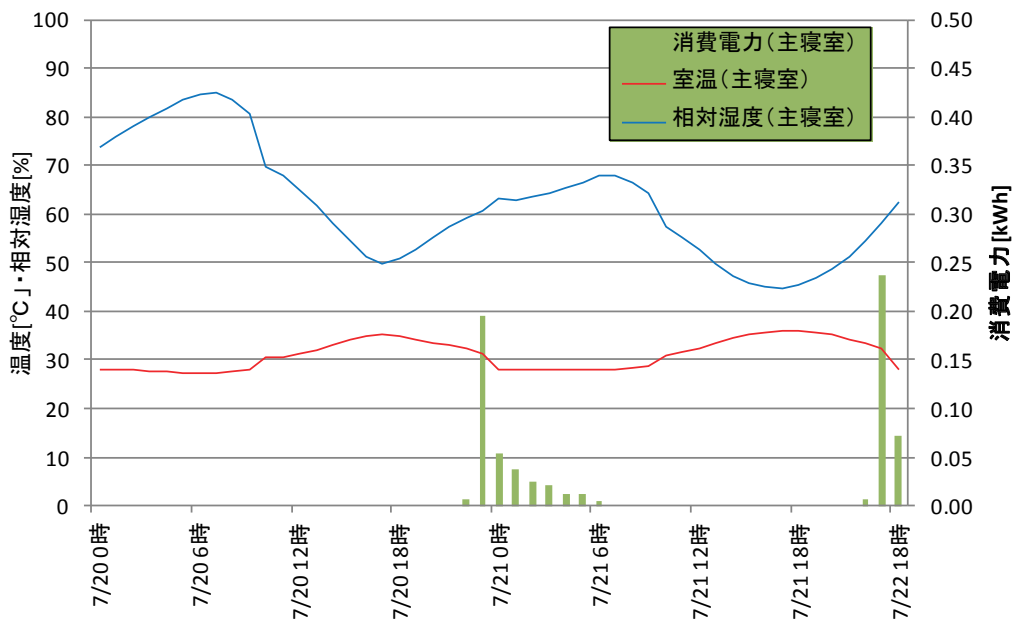


図 292 室温湿度・消費電力量変動 (6 地域、主寝室、室温のみによる ON/OFF 判定)

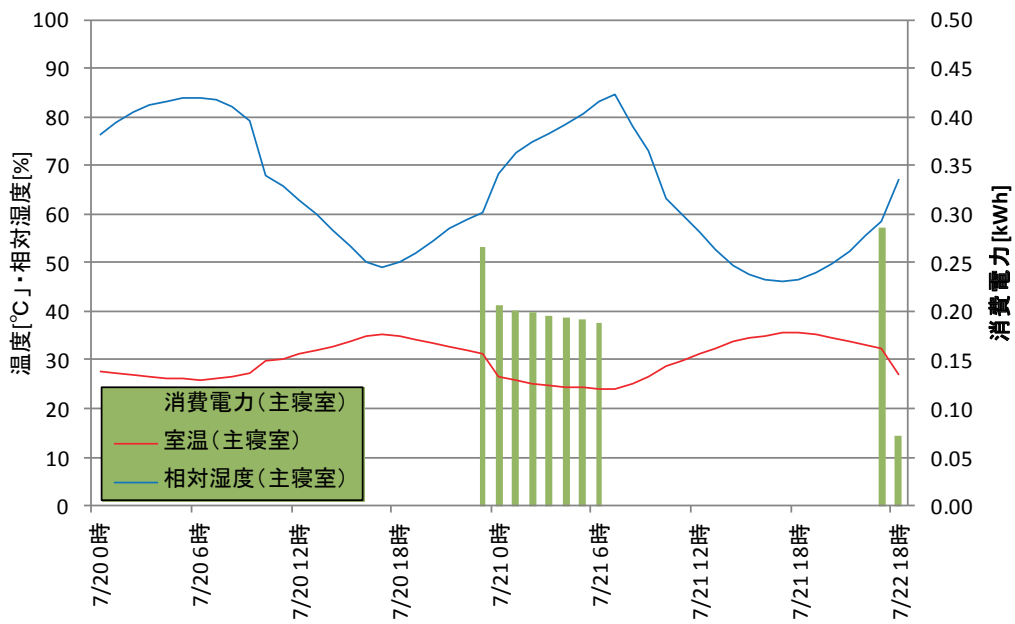


図 293 室温湿度・消費電力量変動 (6 地域、主寝室、人体による ON/OFF 判定)

主寝室の結果では、人体による ON/OFF 判定ではいったん ON になると在室中は常に冷房 ON となるため、7/20 から 21 日にかけて夜間は常にエアコンが稼働し、室温が 25°C 以下まで低下している。その間の潜熱処理量は小さいため、室の相対湿度は大きく上昇している。

一方、室温のみによる ON/OFF 判定では 1 分単位でエアコンの ON/OFF 判定が行われ (28°C を超過しているかどうか)、結果的には夜間の室温がほぼ 28°C に維持されている。

同様に、8 地域における代表的な 2 日間の室温湿度変動およびエアコン消費電力量の変動を示す。

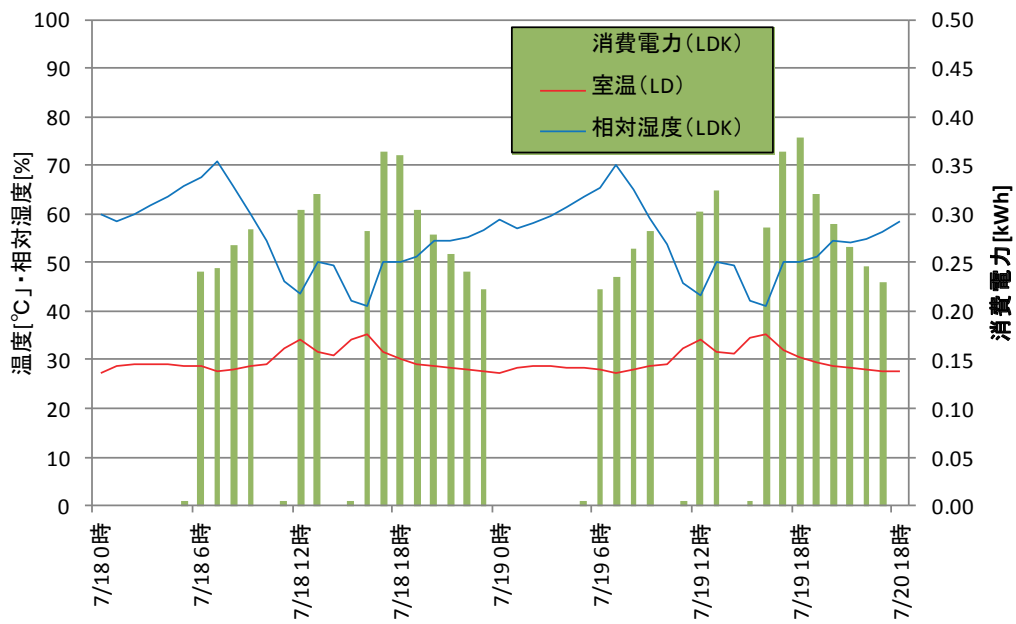


図 294 室温湿度・消費電力量変動（8 地域、LD、室温のみによる ON/OFF 判定）

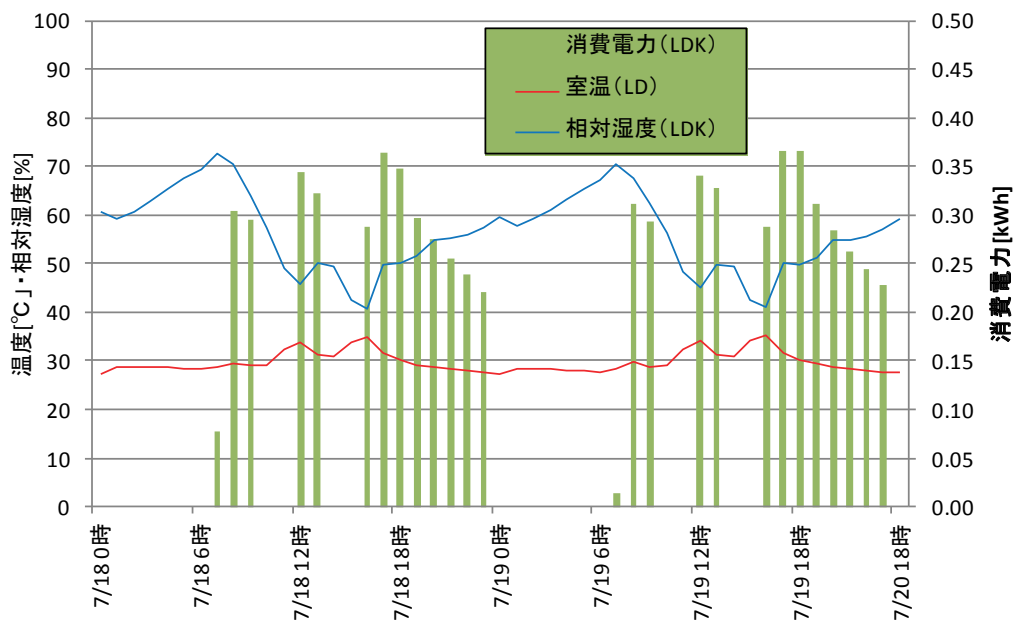


図 295 室温湿度・消費電力量変動（8 地域、LD、人体による ON/OFF 判定）

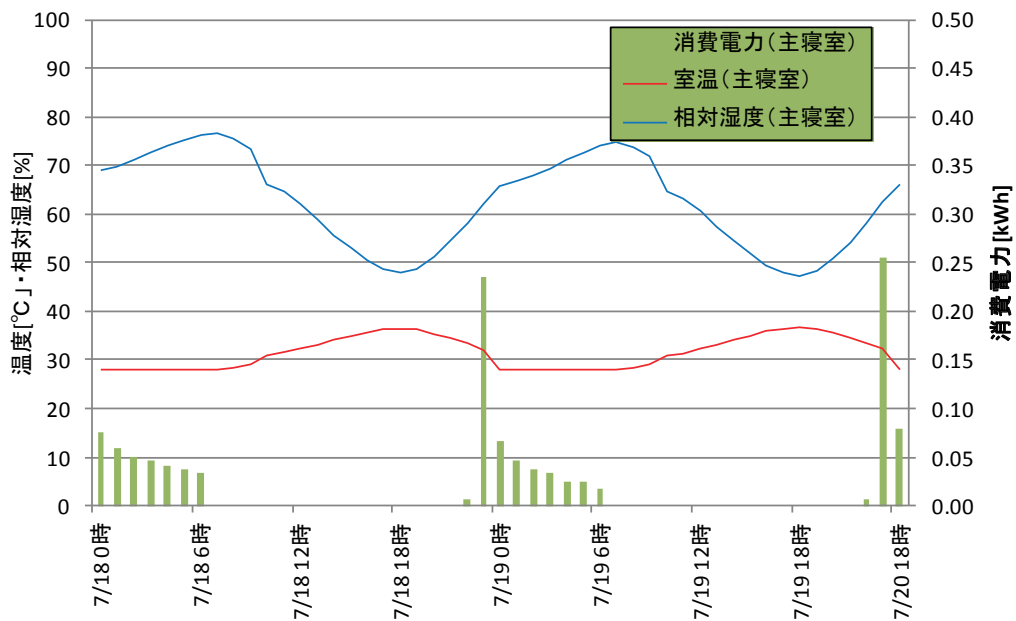


図 296 室温湿度・消費電力量変動（8地域、主寝室、室温のみによる ON/OFF 判定）

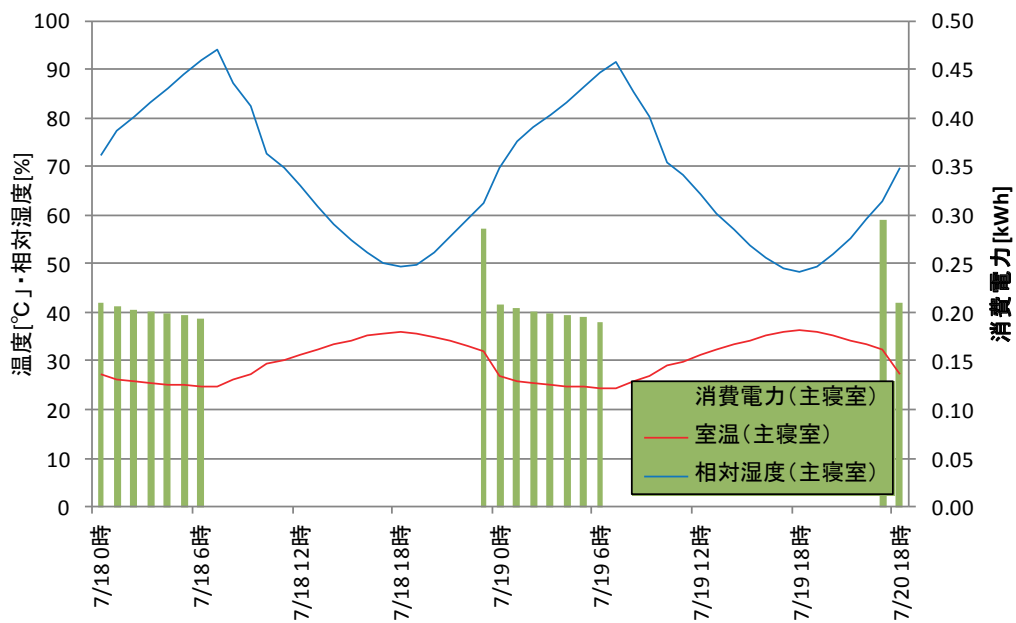


図 297 室温湿度・消費電力量変動（8地域、主寝室、人体による ON/OFF 判定）

計算結果から読み取れる傾向として、

- ・盛夏期の日中の冷房室温は 27℃まで低下することは少ないが、室相対湿度は 50%程度まで低下する。
- ・盛夏期の夜間の冷房室温はおおむね設定温度（28℃）を実現し、人体による判定を考慮した場合はさらに低くなる。ただし潜熱処理量は小さく、60%を実現することはできない。
- ・梅雨時や蒸暑時は、夜間の傾向に近い結果となる。
- ・暑く乾燥している日は顕熱処理量が卓越し、人体判定の有無の影響は小さく、Simheat の結果との差も小さくなると考えられる。

表 141 の値はこれらの傾向が複合的に生じた結果であるが、

- ・室温のみで判定した場合、(基準値との比較で) 寒冷地ほど冷房エネルギーが大きく、蒸暑地では小さくなる・・・SimHeat より潜熱除去量が小さいため
- ・人体で判定した場合、(基準値との比較で) どの地域も平均的に冷房エネルギーが大きい・・・夜間などで室温が低下しても冷房 ON の時間が続くため

という傾向を反映したための結果と考えられる。

6 まとめ

本調査では、建物の熱負荷特に潜熱負荷を適切に評価するため、

- ・建物要因
- ・人体要因
- ・設備要因

の3要因を設定し、それぞれの特性をモデル化するための各種調査等を行い、実際のモデル化（プログラム化）と計算を行った。

建物要因に関しては、アンケートにより住宅における家具・収納物の使用実態調査を行うとともに、家具・収納物の吸放湿特性の文献調査、熱湿気モデルの作成を行った（第2章）。

人体要因に関しては、アンケート調査結果を踏まえてエアコンの使用実態を調査・整理し、窓の開閉などを含めて外気温・室温・在室状況に基づいて冷房行為を決定するモデルを作成した（第4章）。

設備要因に関しては、非住宅・住宅ともに実物件における実測調査および実験室実験を行い、ビル用マルチ、調湿換気装置、全熱交換器、エアコンの各設備の特性を把握した。これらの顕熱・潜熱処理量に関するモデルを作成して実データとの比較検討を行い、おおむね良好な結果が得られた（第3章）。

これらの各要因のモデルをまとめたプログラムを作成し、住宅の冷房一次エネルギーの基準値に相当する値が得られた（第5章）。

参考文献

- i 西岡真稔・松尾陽・井上隆・金成完、紙の湿気特性に関する研究 (その1) —紙の吸放湿特性—、日本建築学会大会学術講演梗概集、pp.945-946、1987
- ii 金成完・松尾陽・井上隆・西岡真稔、紙の湿気特性に関する研究 (その2) —湿気伝導率の推定—、日本建築学会大会学術講演梗概集、pp.947-948、1987
- iii 松尾陽、室の潜熱容量について、日本建築学会大会学術講演梗概集、pp.815-816、1987
- iv 金成完・松尾陽・井上隆・西岡真稔、紙の湿気特性に関する研究 (その3) 書類の吸湿特性、日本建築学会大会学術講演梗概集、pp.565-566、1988
- v 金桓龍・松尾陽・永田明寛、室内潜熱負荷と湿度変動に関する研究 —繊維類の湿気特性—、日本建築学会大会学術講演梗概集、pp.1225-1226、1993
- vi 金桓龍・松尾陽・坂本雄三・永田明寛、潜熱負荷と湿度変動に及ぼす室内備品類の吸放湿性の影響に関する研究 その1 寝具類の湿気物性値の測定、日本建築学会大会学術講演梗概集、pp.315-316、1996
- vii 坂本雄三・松尾陽・永田明寛・金桓龍、潜熱負荷と湿度変動に及ぼす室内備品類の吸放湿性の影響に関する研究 その2 シミュレーションによる吸放湿性の影響に関する検討、日本建築学会大会学術講演梗概集、pp.317-318、1996
- viii 出端裕輔・梅野徹也、家具の吸放湿性が室内温湿度に及ぼす影響についての検討、日本建築学会大会学術講演梗概集、pp.267-268、2012
- ix 梅野徹也・出端裕輔、室内衣類・家具の吸放湿効果の評価、日本建築学会環境工学委員会第44回熱シンポジウム、2014
- x 松本衛：新建築学体系 10 環境物理、湿気、彰国社刊、pp.131~133、1984
- xi 池田哲朗：結露障害防止に関する基礎的研究、京都大学博士論文、1980
- xii 梅野徹也・出端裕輔、室内衣類・家具の吸放湿効果の評価、日本建築学会環境工学委員会第44回熱シンポジウム、2014
- xiii 一般財団法人 建築環境・省エネルギー機構：平成25年省エネルギー基準に準拠した算定・判断の方法及び解説 II 住宅、2013.5
- xiv Leon R. Glicksman and Steven Taub: Thermal and behavioral modeling of occupant-controlled heating, ventilating and air conditioning systems, *Energy and Buildings*, Vol.25 (3), pp.243-249, 1997
- xv H.B. Rijal, P. Tuohy, M.A. Humphreys, J.F. Nicol, A. Samuel and J. Clarke: Using results from field surveys to predict the effect of open windows on thermal comfort and energy use in buildings, *Energy and Buildings*, Vol.39 (7), pp.823-836, 2007.7
- xvi Sebastian Herkel, Ulla Knapp and Jens Pfafferott: Towards a model of user behaviour regarding the manual control of windows in office buildings, *Building and Environment*, 43 (4), pp.588-600, 2008.4
- xvii Geun Young Yun and Koen Steemers: Time-dependent occupant behaviour models of window control in summer, *Building and Environment*, Vol.43 (9), pp.1471-1482, 2008.9
- xviii Frederic Haldi and Darren Robinson: On the behaviour and adaptation of office occupants, *Building and Environment*, Vol.43 (12), pp.2163-2177, 2008.12
- xix Geun Young Yun, Pal Tuohy and Koen Steemers: Thermal performance of a naturally ventilated building using a combined algorithm of probabilistic occupant behaviour and deterministic heat and mass balance models, *Energy and Buildings*, Vol.41 (5), pp.489-499, 2009.5
- xx Yufan Zhang and Peter Barrett: Factors influencing the occupants' window opening behaviour in a naturally ventilated office building, *Building and Environment*, Vol.50, pp.125-134, 2012.4
- xxi 谷本潤、萩島理、片山忠久：冷房発停にマルコフモデルを適用した場合の熱負荷特性に関する基礎的検討：第3報-実測に基づく冷房発停状態の遷移特性に関する考察、空気調和・衛生工学会論文集、

No.82, pp.59-66, 2001.7

xxii 羽原宏美、鳴海大典、下田吉之、水野稔：居住者の室内温熱環境調節行為のモデル化による住宅の空調エネルギー消費の予測、人間と生活環境、Vo.11 (2)、pp.83-88、2005.11

xxiii Frederic Haldi: A probabilistic model to predict building occupants' diversity towards thermal interactions with the building envelope, Proceedings of the 12th Conference of The International Building Performance Simulation Association, 2013.11

xxiv Rune Andersen, Valentina Fabi, Jorn Toftum, Stefano P. Corgnati and Bjarne W. Olesen: Window opening behaviour modelled from measurements in Danish dwellings, Building and Environment, Vol.69, pp.101-113, 2013.11

xxv 羽原宏美、鳴海大典、下田吉之、水野稔：一般住戸を対象とした実態調査に基づく冷房発停の生起要因に関する検討、日本建築学会環境系論文集、No.589、pp.83-90、2005.3

xxvi 澤地孝男、松尾陽、羽田野健、福島弘幸：暖冷房行為生起の決定要因と許容室温範囲に関する検討 住宅の室内気候形成に寄与する居住者の行動に関する研究 その1、日本建築学会計画系論文報告集、No.382、pp.48-59、1987.12

xxvii 菅原正則、本間博文、梅干野晁、飯野由香利：夏季の自然通風時における住居平面と住まい方の室内気候への影響に関する調査研究、空気調和・衛生工学会論文集、No.57、pp.117-127、1995.2

xxviii 鈴木憲三、松原斎樹、森田大、澤地孝男、坊垣和明：札幌、京都、那覇の公営集合住宅における暖冷房環境の比較分析、暖冷房使用に関する意識と住まい方の地域特性と省エネルギー対策の研究 その1、日本建築学会計画系論文集、No.475、pp.17-24、1995.9

xxix 垂水弘夫、久保猛志、酒井健興：北陸の戸建住宅における温冷感を中心とした居住者意識調査、断熱仕様・暖冷房等の実態と快適性評価の高い住宅の抽出、日本建築学会計画系論文集、No.488、pp.25-34、1996.10

xxx 前真之、鍋島美奈子、永村一雄、長井達夫、二宮秀興、鎌田元康：集合住宅における冷房の使用状況に関する研究、空気調和・衛生工学会論文集、No.87、pp.41-50、2002.10

xxxi 澤島智明、松原斎樹、藏澄美仁：防暑行為の実施実態と居住者の意識・価値観、プレハブ住宅居住者の夏期の防暑行為に関する研究 その1、日本建築学会環境系論文集、No.578、pp.9-15、2004.4

xxxii 飛田国人、松原斎樹、井上ともみ、谷村真由美、青地奈央、須藤由佳子、地濃祐介、藏澄美仁：京都市の集合住宅居住者の夏期の室内熱環境調節手法における理想と実態の関係、日本建築学会環境系論文集、No.625、pp.385-391、2008.3

xxxiii 小林茂雄、村中美奈子、中山和美：夏期の空調行為に与える周辺環境条件の影響の個人差 仮想環境に対する行為の推定結果を基に、日本建築学会環境系論文集、No.631、pp.1147-1153、2008.9

xxxiv 宮田希、松原斎樹、大和義昭、澤島智明、合掌頭、藏澄美仁、飛田国人：夏の涼のとり方に影響する要因の考察、西日本4地域における実態調査より、日本生気象学会雑誌、Vol.49、No.1、pp.23-30、2012.3

xxxv 坪田祐二、中山和美、間純一：夏期の住宅における窓の開放に関する調査研究、首都圏におけるインターネットアンケート調査による窓の開放の意識と実際の窓の開放、日本建築学会環境系論文集、No.676、pp.515-521、2012.6

xxxvi 澤島智明、松原斎樹：居住者の冷房の仕方と暖房の仕方の関係、プレハブ住宅居住者の夏期・冬期の環境調節行為に関する研究、日本建築学会環境系論文集、No.695号、pp.27-35、2014.1

xxxvii 総務省統計局：平成22年国勢調査人口等基本集計、2011.10

xxxviii 福島逸成、林徹夫、浦野良美、龍有二、渡辺俊行、赤司泰義：福岡における夏季の住まい方と住宅の冷房エネルギー消費量に関する研究、空気調和・衛生工学会論文集、No.61、pp.79-90、1996.4

xxxix 坊垣和明、澤地孝男、吉野博、鈴木憲三、赤林伸一、井上隆、大野秀夫、松原斎樹、林徹夫、森田大：全国的調査に基づく住宅の暖冷房時間および暖冷房期間に関する研究、日本建築学会計画系論文集、No.509、pp.41-47、1998.7

xl 梅宮典子、吉田治典、小林恒夫：冷房利用時間長さの要因に関する研究：京都における大学生の冷房利用行動について、日本建築学会計画系論文集、No.543、pp.101-108、2001.5

xli 浅輪貴史、梅干野晁、武澤秀幸、清水敬示：戸建住宅における窓開閉・冷房使用の行動特性と影響要因解析、屋外空間の微気候と居住者の開放的な住まい方との関わりに関する研究 その2、日本建築

学会環境系論文集、Vol. 593、pp.87-94、2005.7

^{xlii} シュバイカ マーセル、宿谷昌則：夏季夜間におけるエアコン使用の個人差に関する調査、日本建築学会環境系論文集、Vol.73、No.633、pp.1275-1282、2008.11

^{xliii} Frederic Haldi, Darren Robinson: Interactions with window openings by office occupants, *Building and Environment*, Vol.44, pp.2378-2395, 2009.12

^{xliv} Frederic Haldi, Darren Robinson: On the unification of thermal perception and adaptive actions, *Building and Environment*, Vol.45, pp.2440-2457, 2010.12

^{xlv} Shen Wei, Richard Buswell, Dennis Loveday: Factors affecting 'end-of-day' window position in a non-air-conditioned office building, *Building and Environment*, Vol.62, pp.87-96, 2013.11

^{xlvi} Nan Li, Juncheng Li, Ruijuan Fan, Hongyuan Jia: Probability of occupant operation of windows during transition seasons in office buildings, *Renewable Energy*, 2014 (in press)

^{xlvii} Xiaoxin Ren, Da Yan, Chuang Wang: Air-conditioning usage conditional probability model for residential buildings, *Building and Environment*, Vol.81, pp.172-182, 2014.11

付録 A

東京 A ビルにおける

ヒートポンプデシカント調湿外調機の性能特性式から計算される

消費電力、全熱能力、潜熱能力と実測値の比較データ

(夏季)

データは、日毎の機器の運転状況（消費電力量（実測値、特性式算出値）、全熱能力（実測値、特性式算出値）、潜熱能力、COP）の1時間毎の平均値および室内、室外、給気の空気状態の1時間毎の平均値を示す。

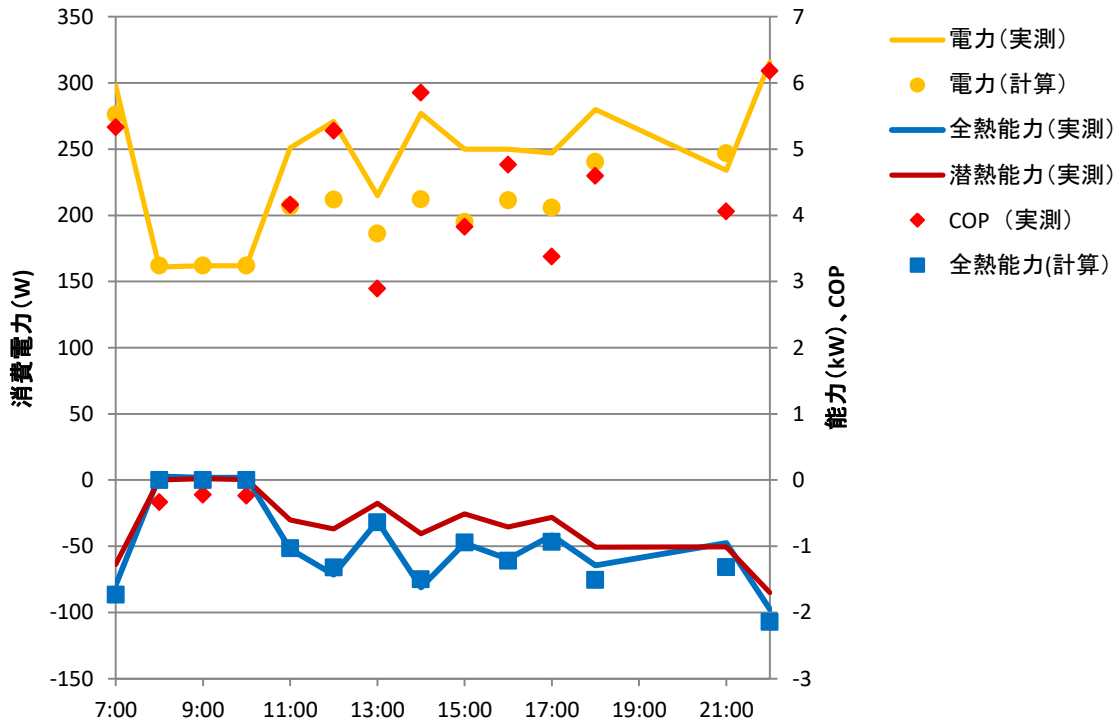


図 1 6/2 運転状況

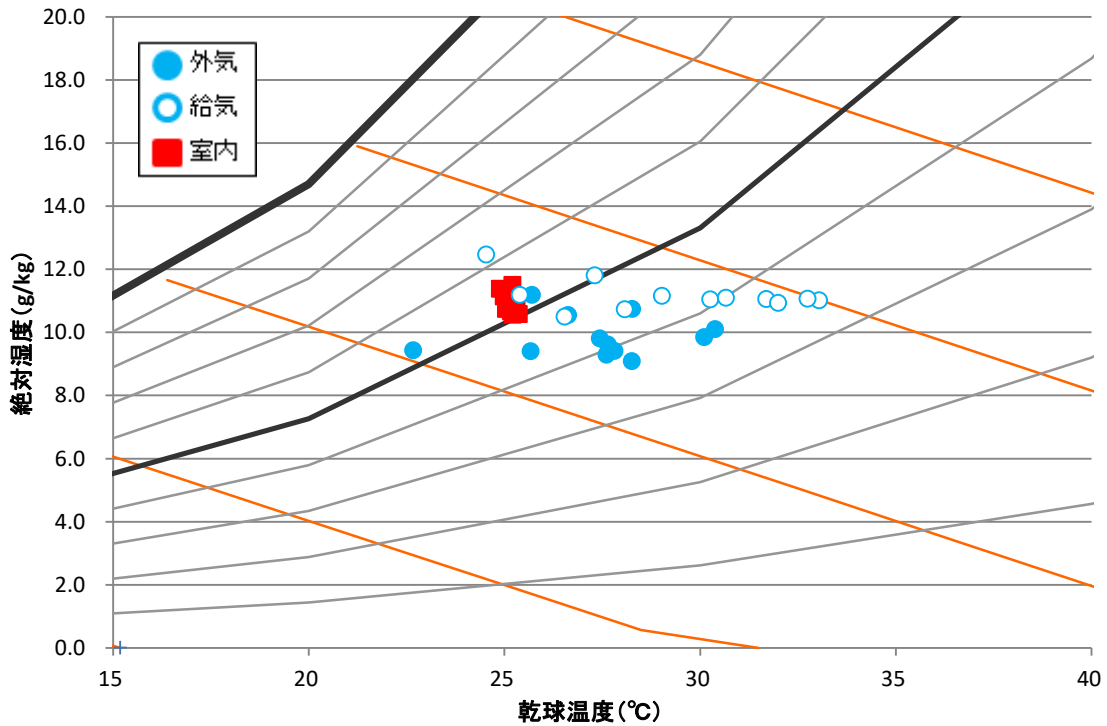


図 2 6/2 空気状態

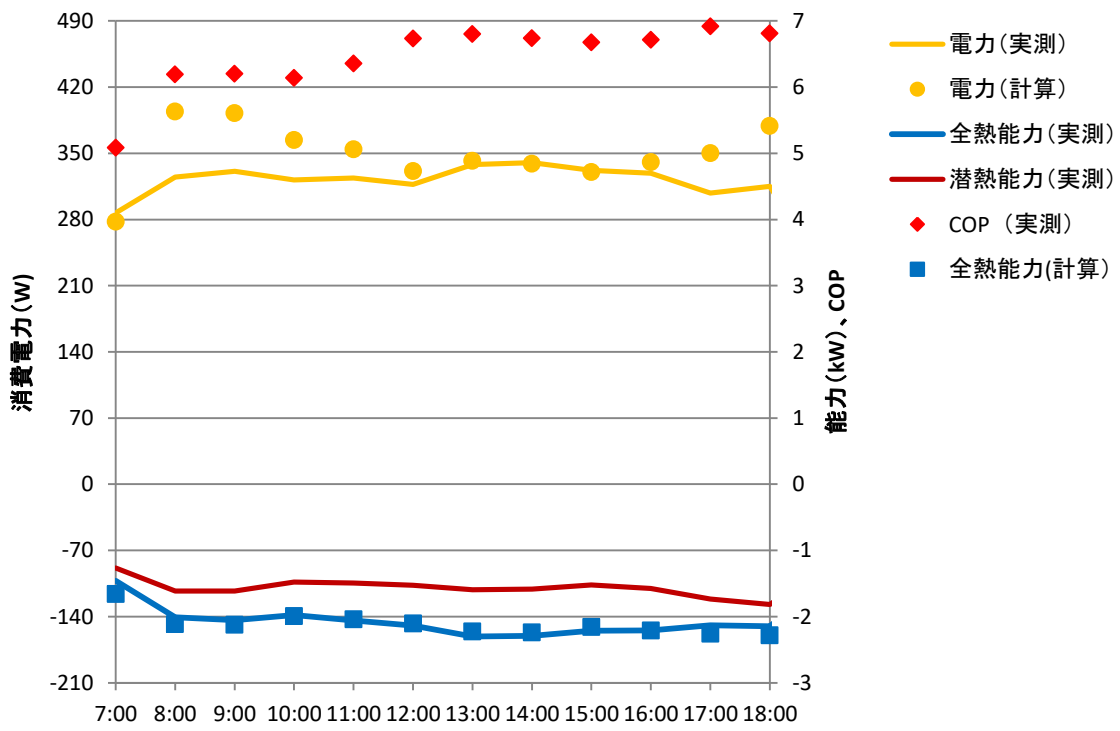


図 3 6/3 運転状況

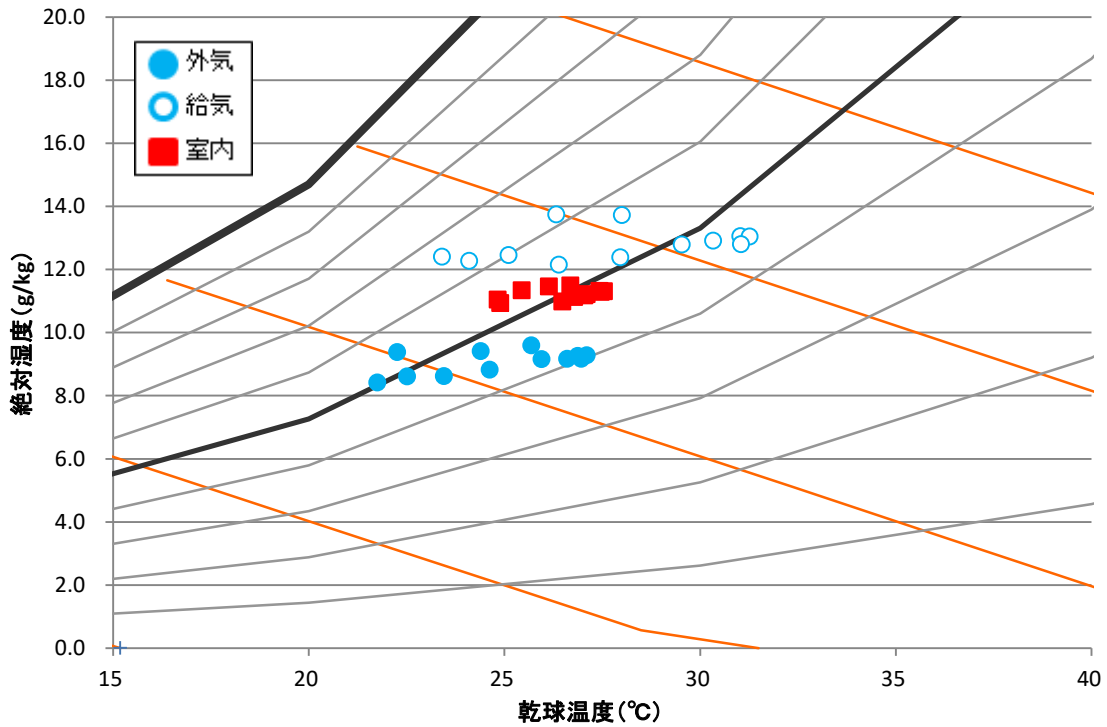


図 4 6/3 空気状態

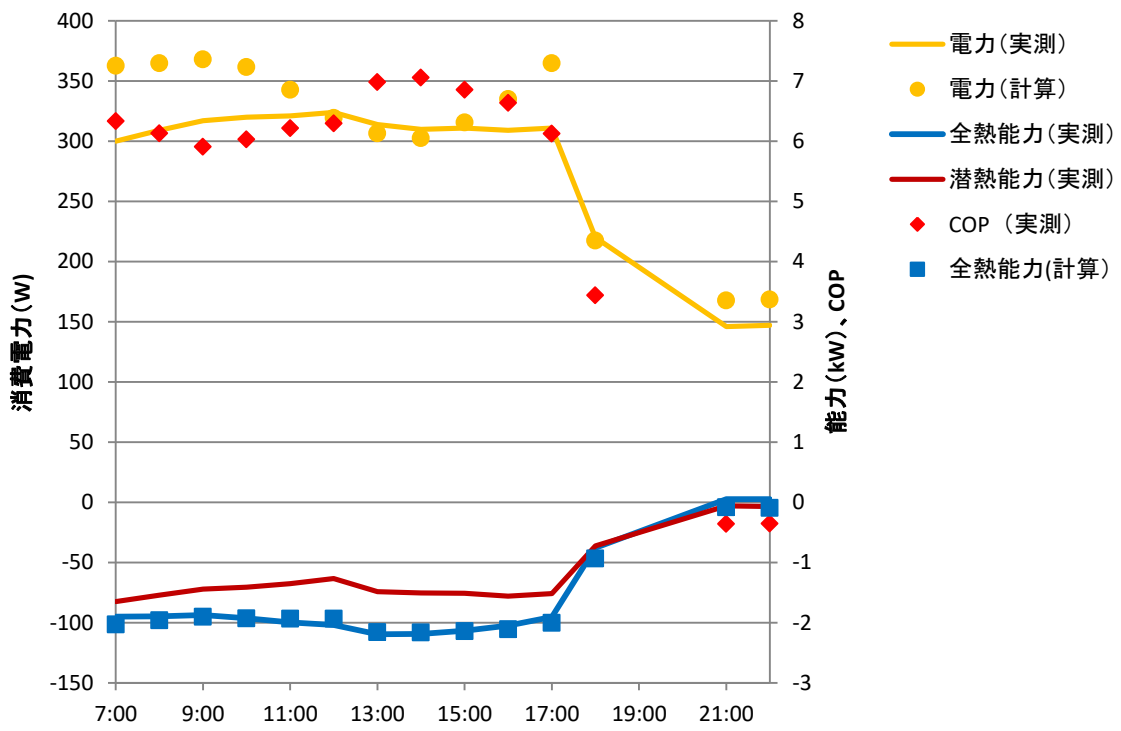


図 5 6/4 運転状態

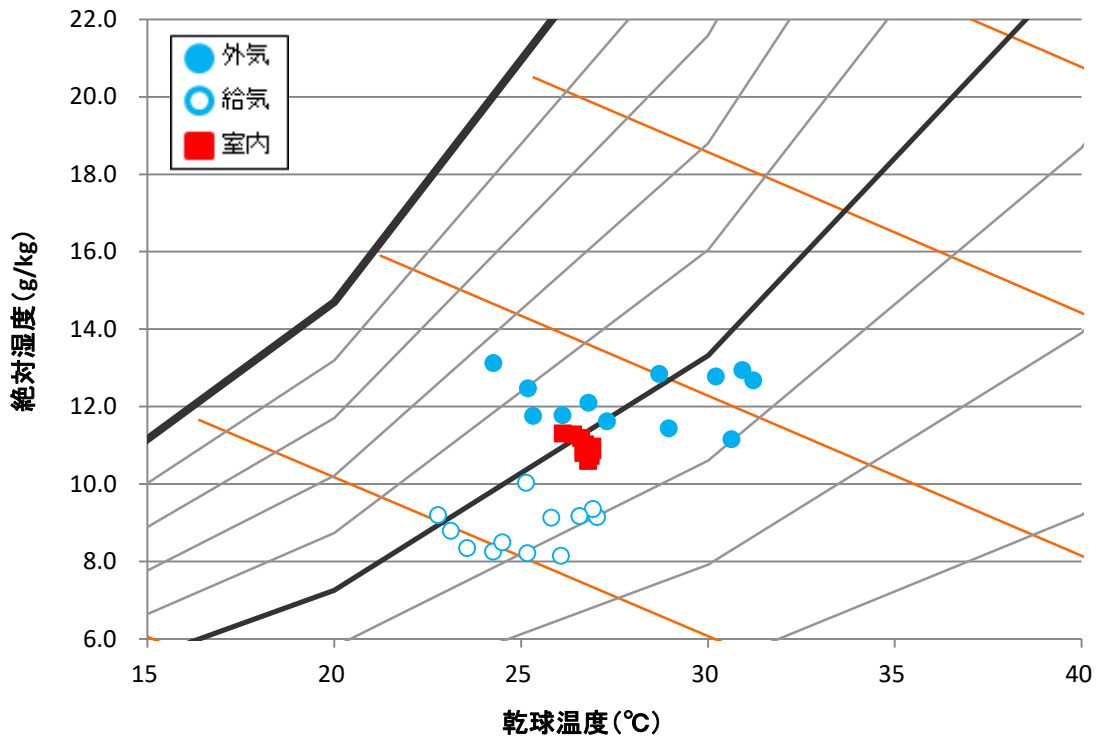


図 6 6/4 空気状態

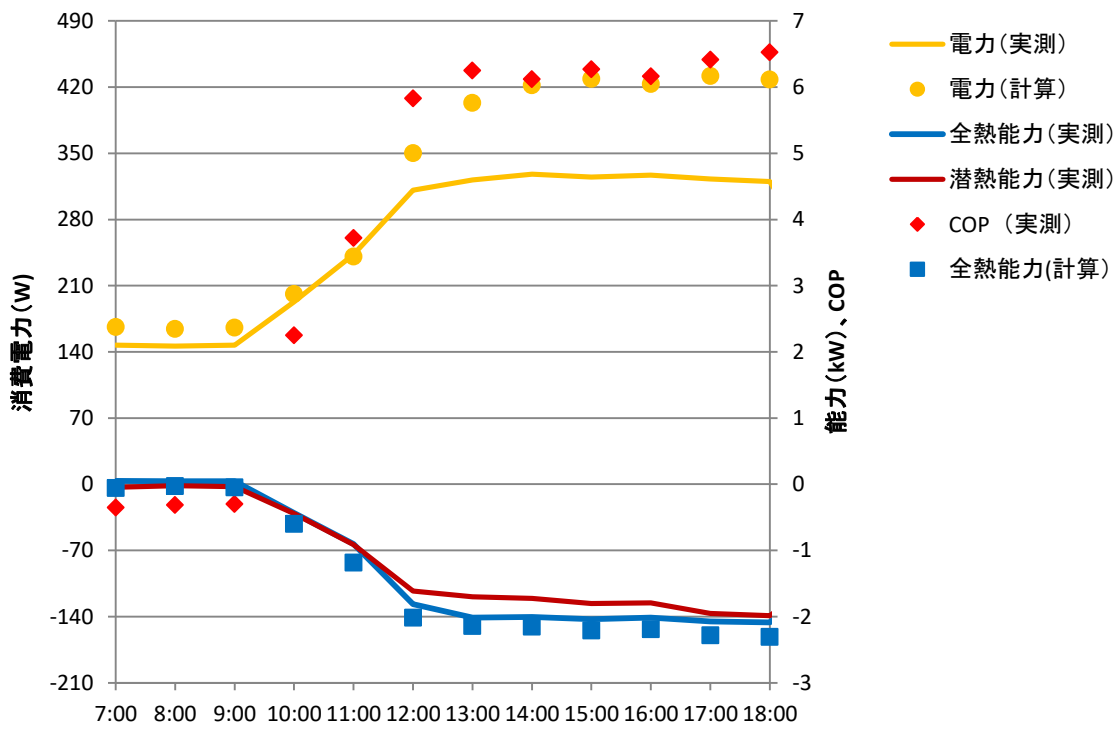


図 7 6/5 運転状態

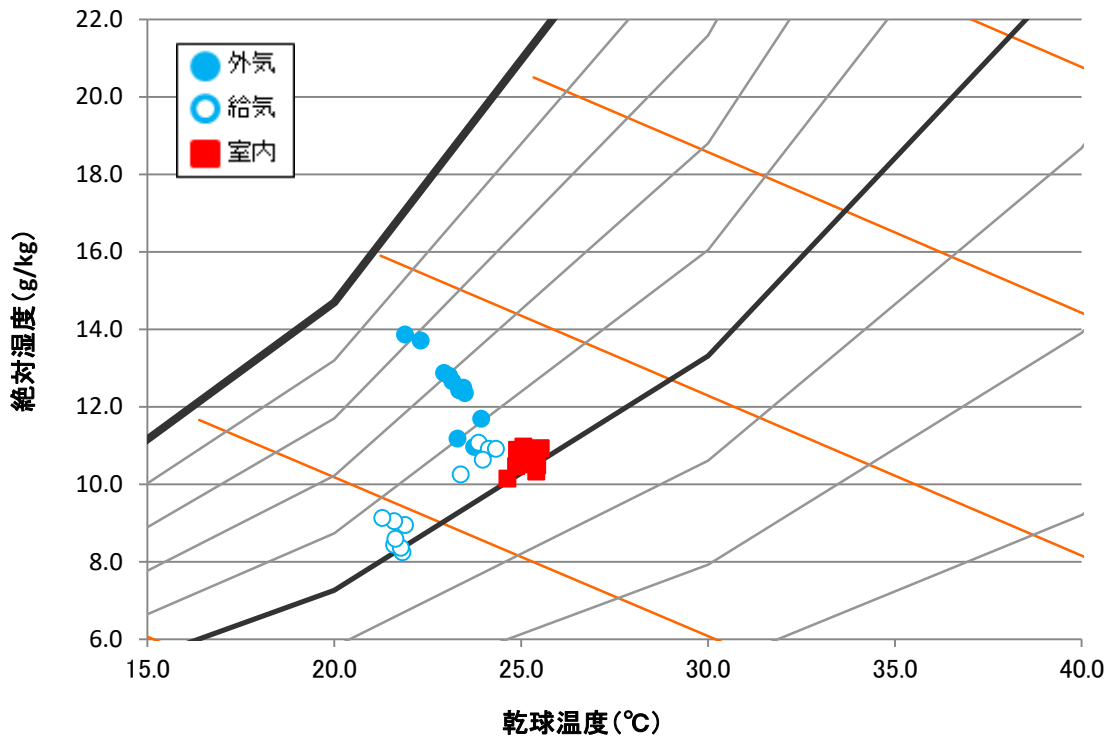


図 8 6/5 空気状態

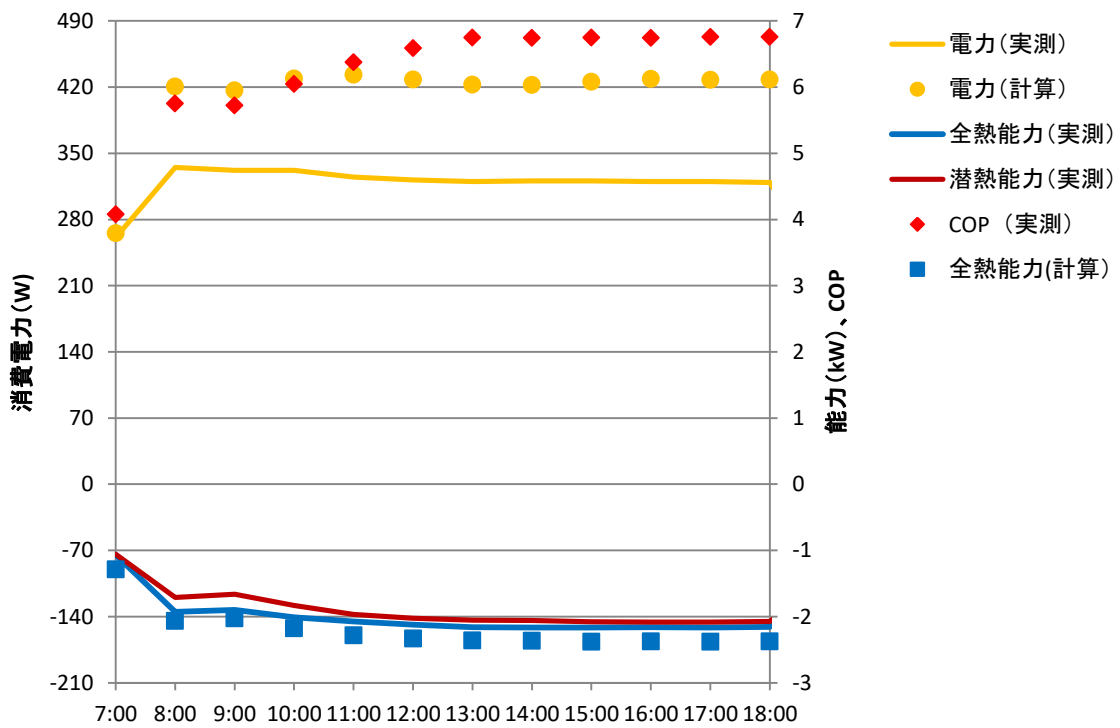


図 9 6/6 運転状態

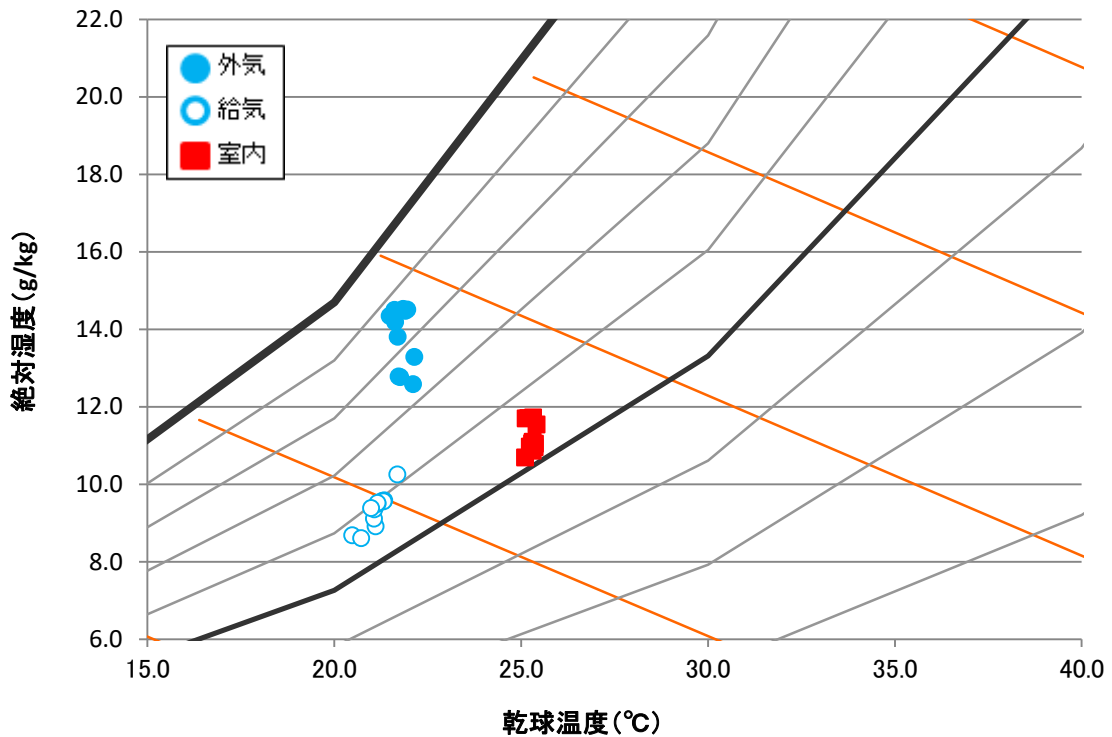


図 10 6/6 空気状態

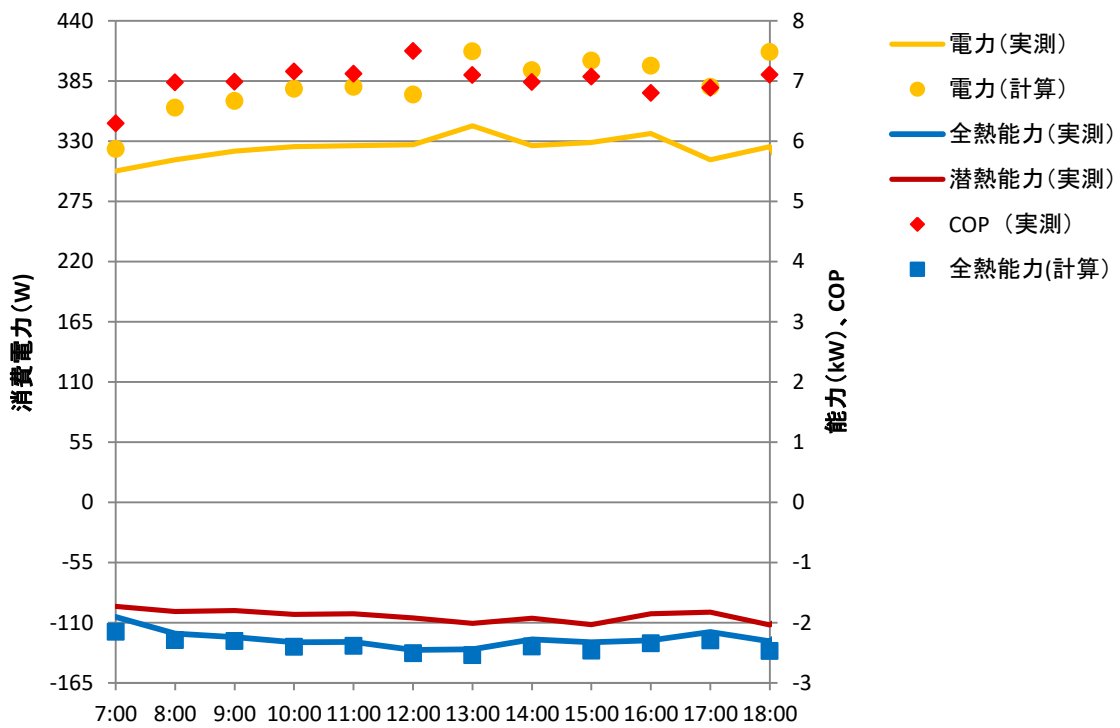


図 11 6/9 運転状態

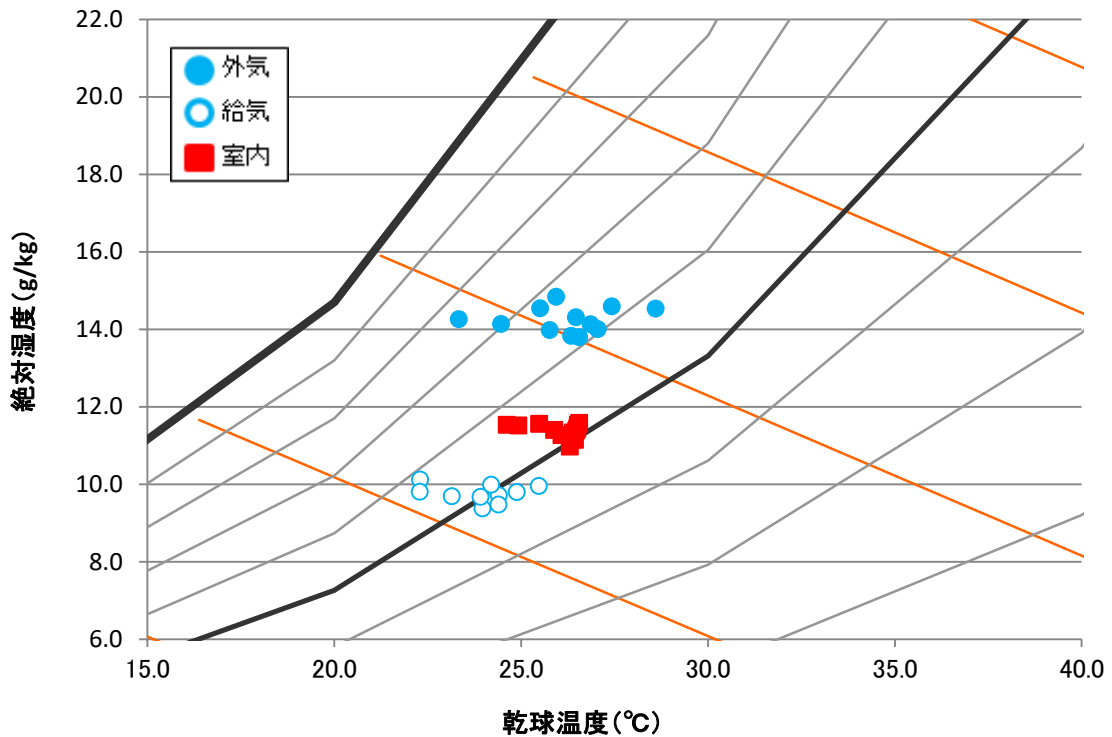


図 12 6/9 空気状態

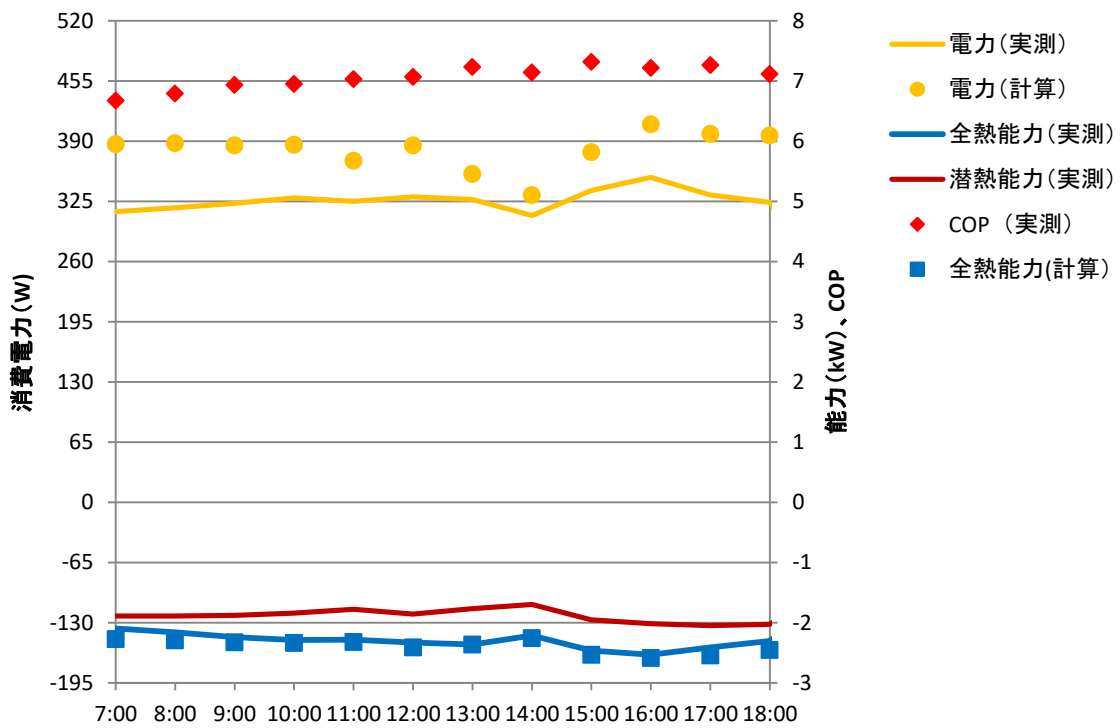


図 13 6/10 運転状態

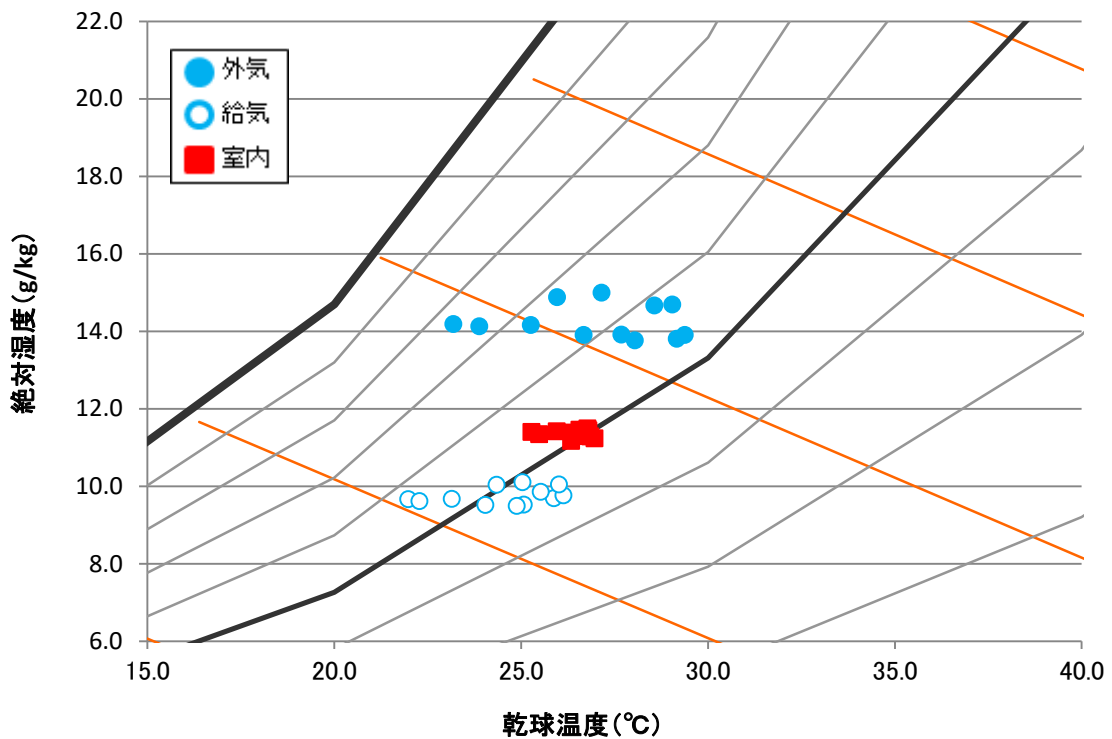


図 14 6/10 空気状態

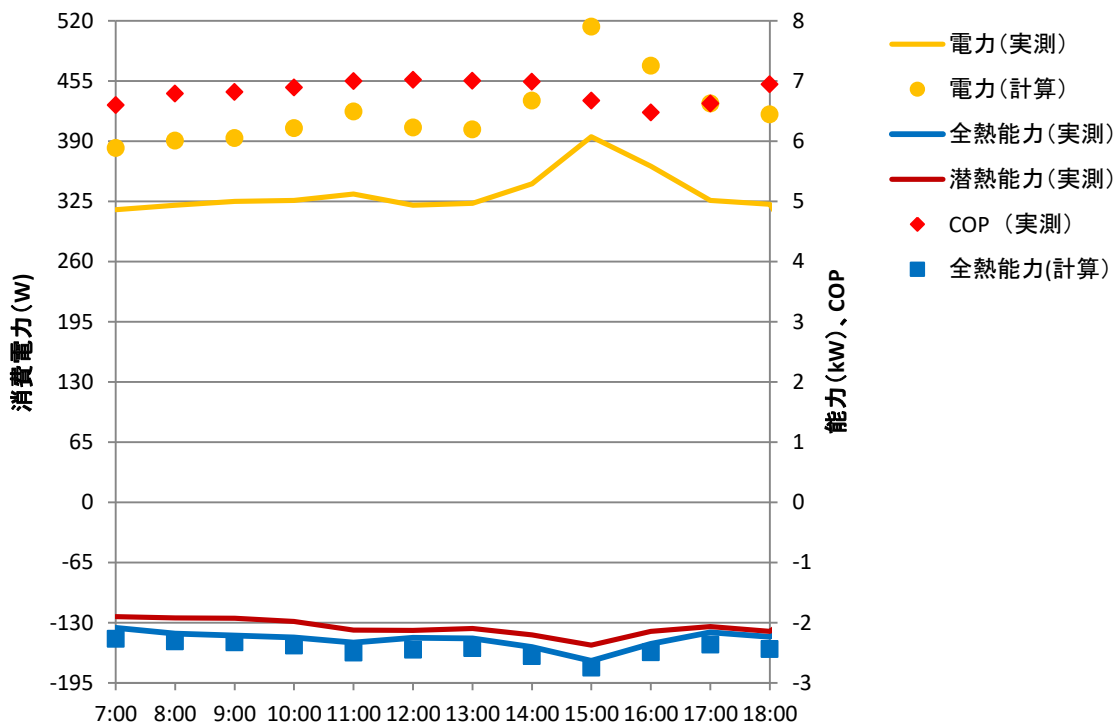


图 15 6/11 运转状态

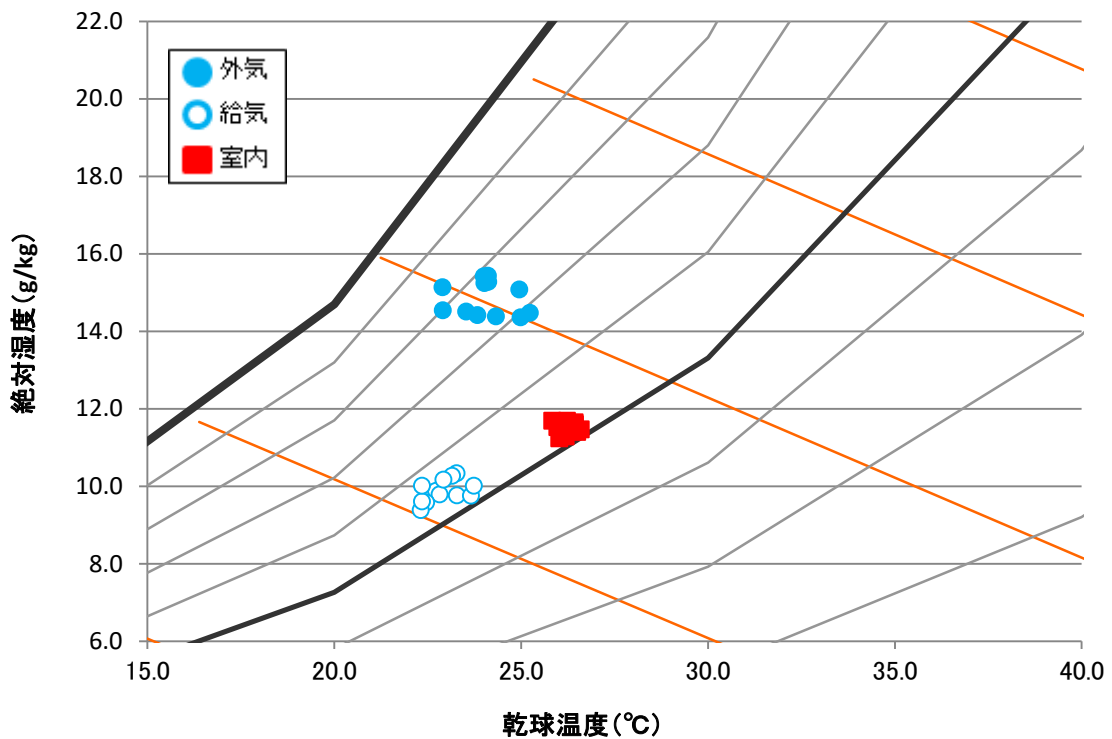


图 16 6/11 空气状态

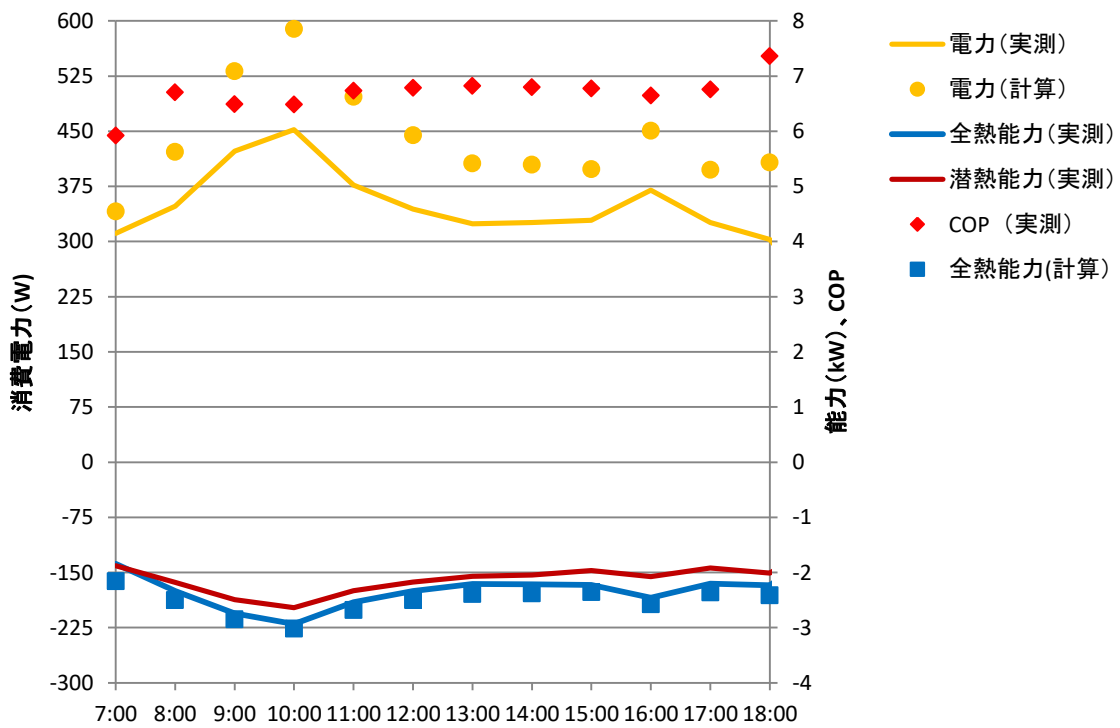


図 17 6/12 運転状態

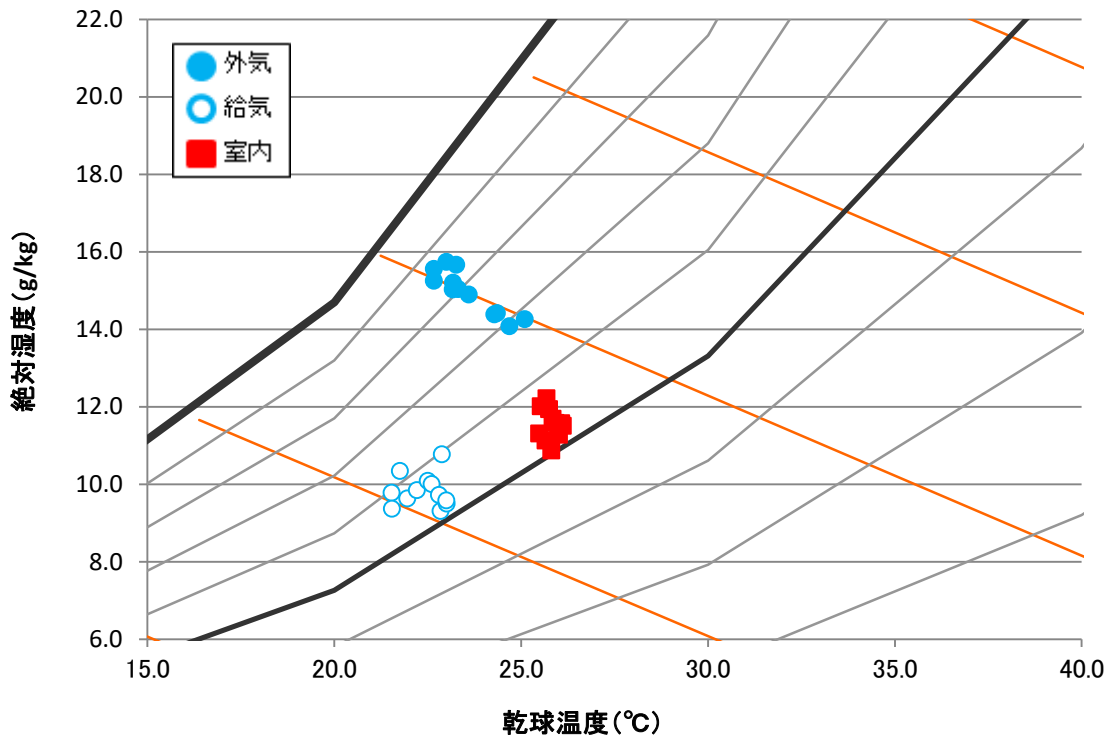


図 18 6/12 空気状態

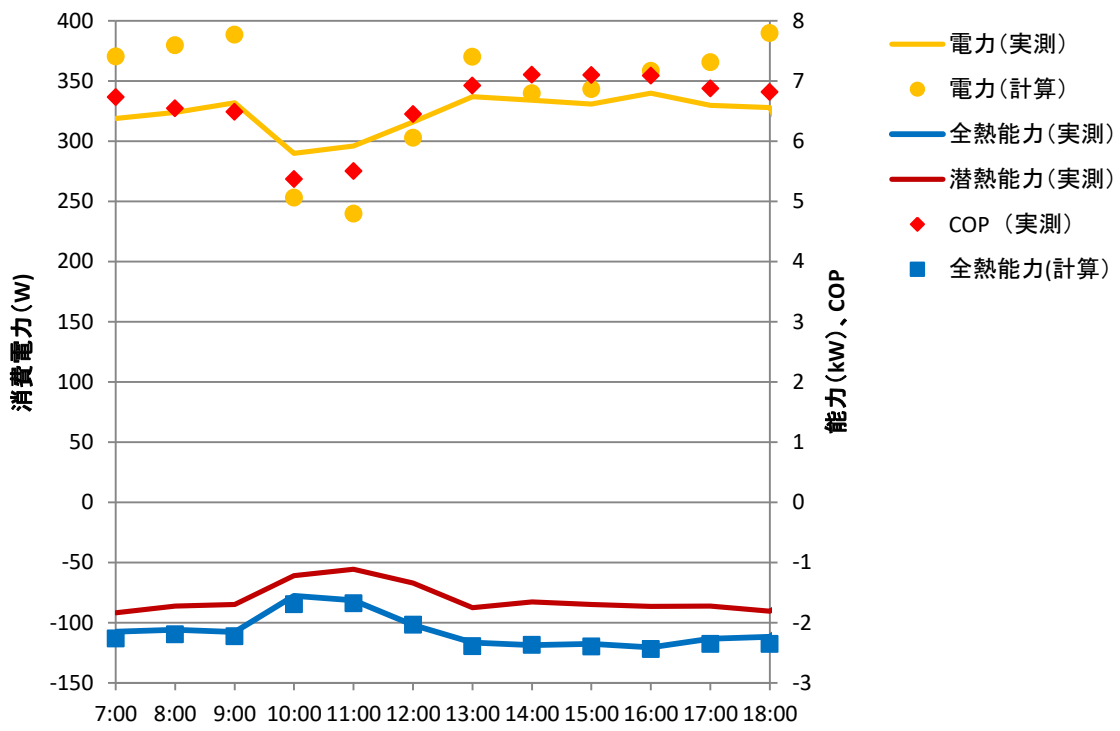


図 19 6/13 運転状態

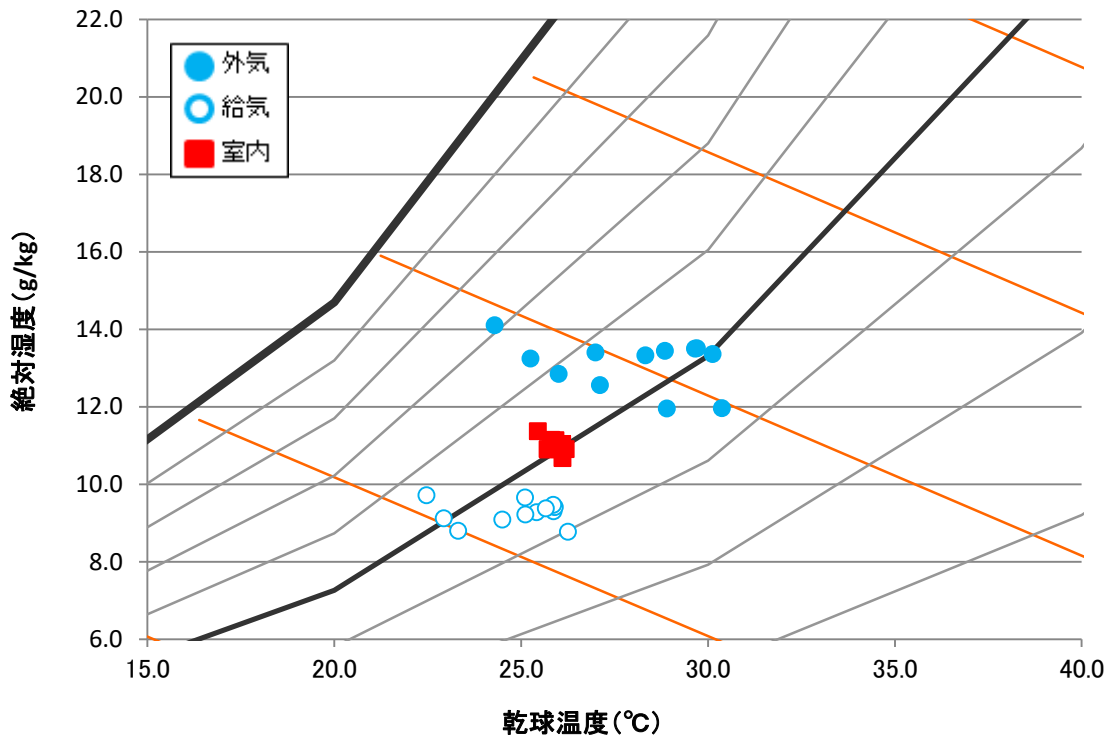


図 20 6/13 空気状態

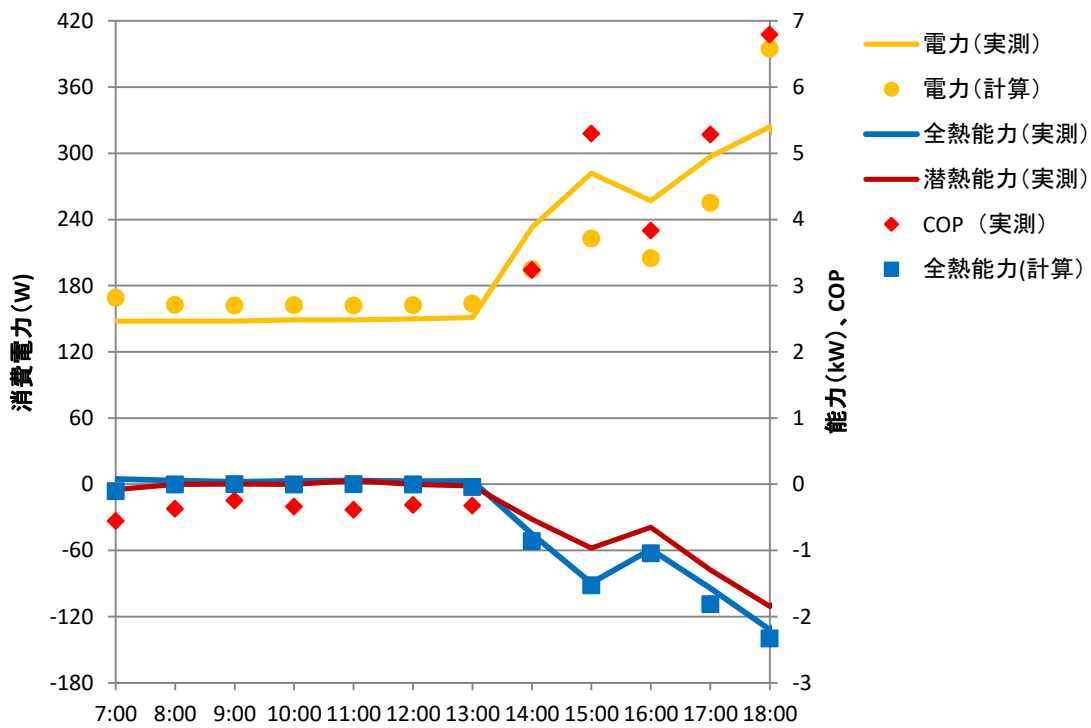


図 21 6/16 運転状態

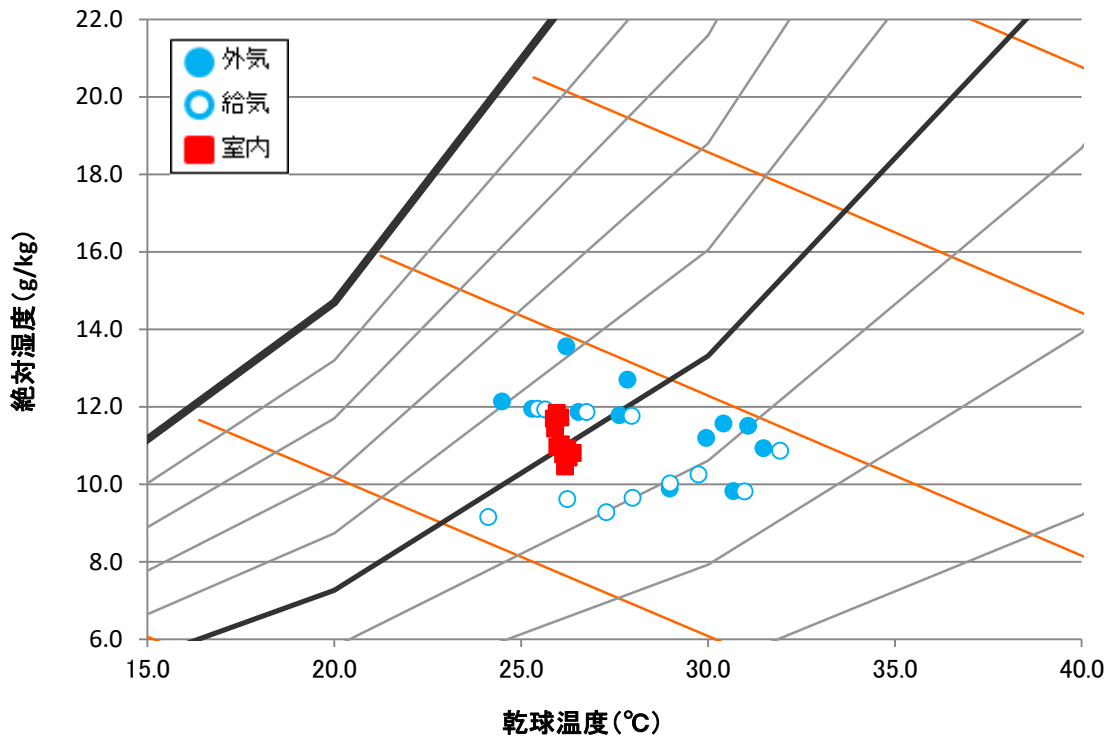


図 22 6/16 空気状態

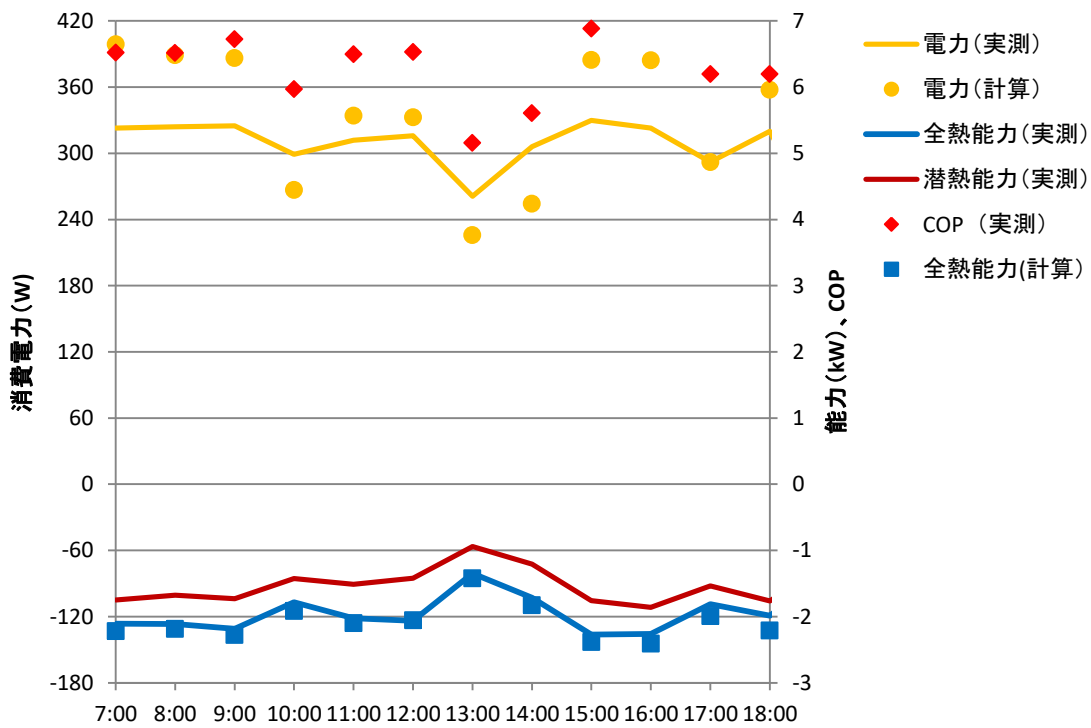


図 23 6/17 運転状態

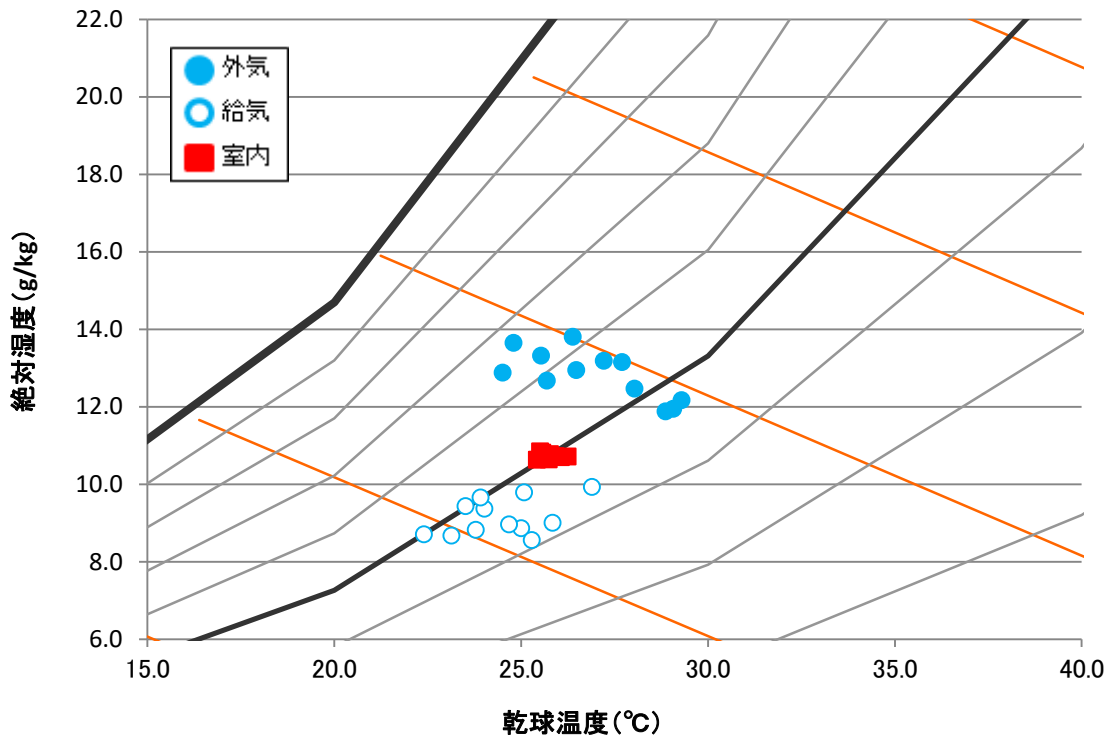


図 24 6/17 空気状態

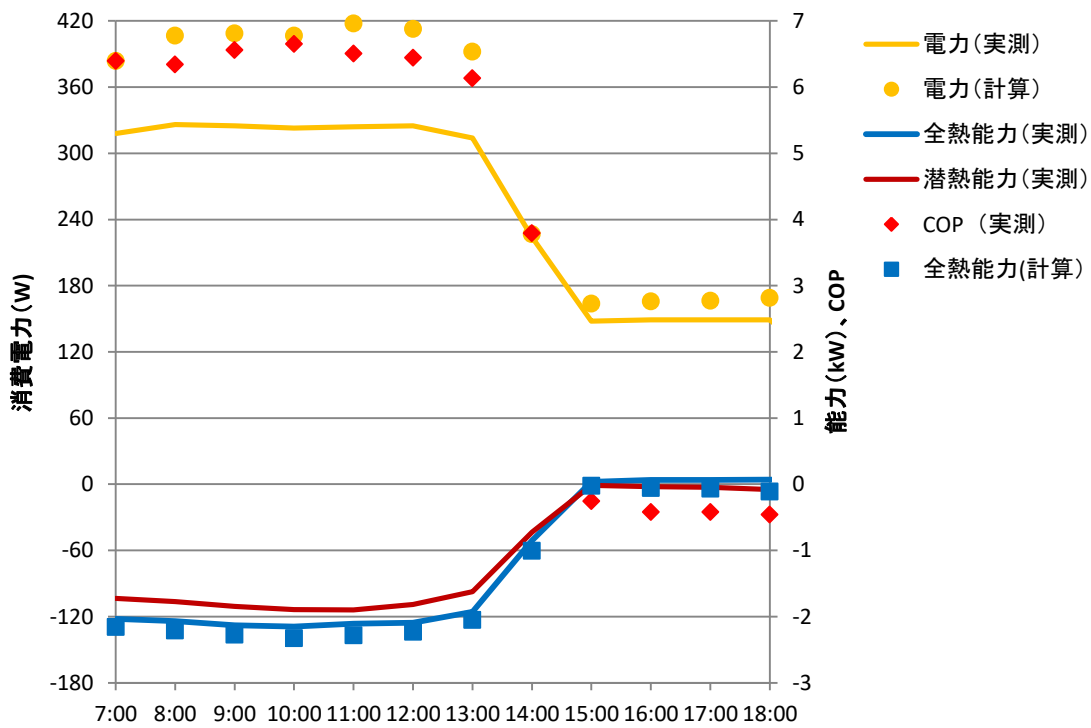


図 25 6/18 運転状態

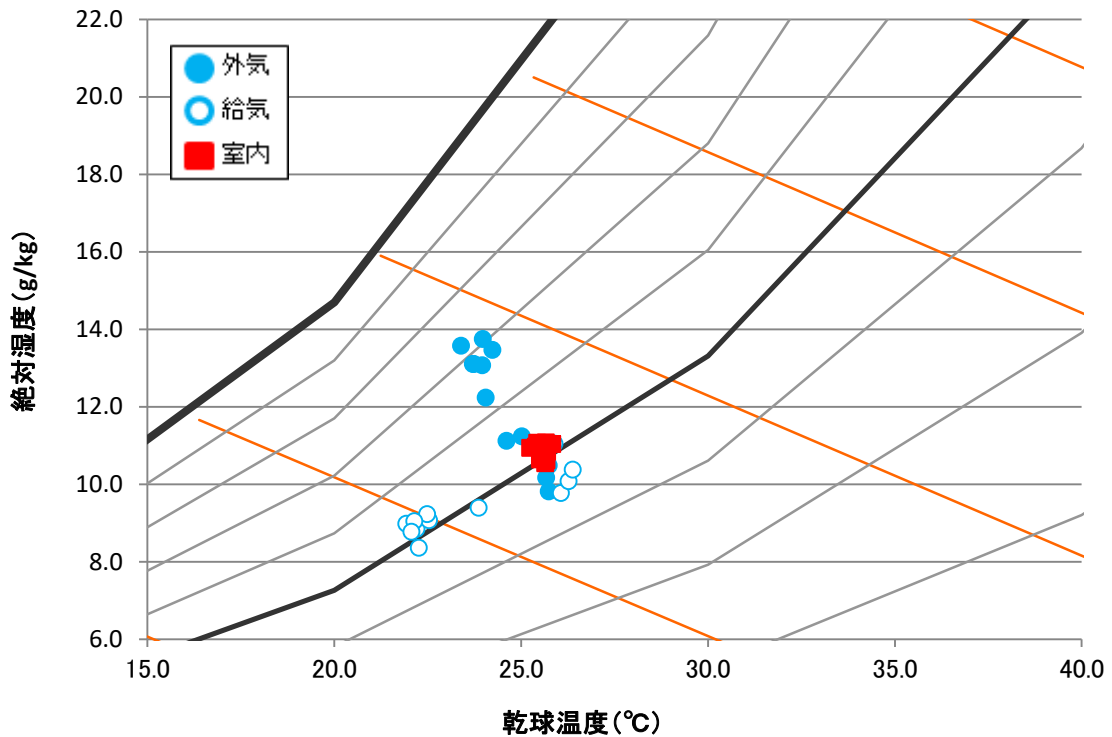


図 26 6/18 空気状態

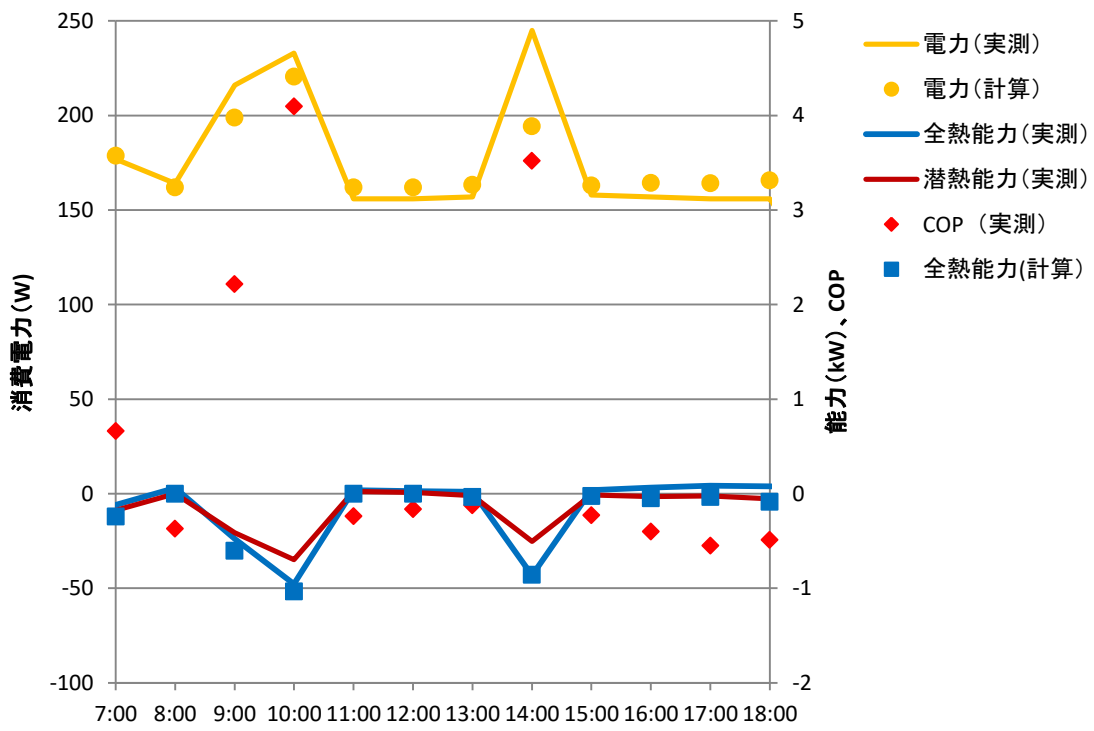


図 27 6/19 運転状態

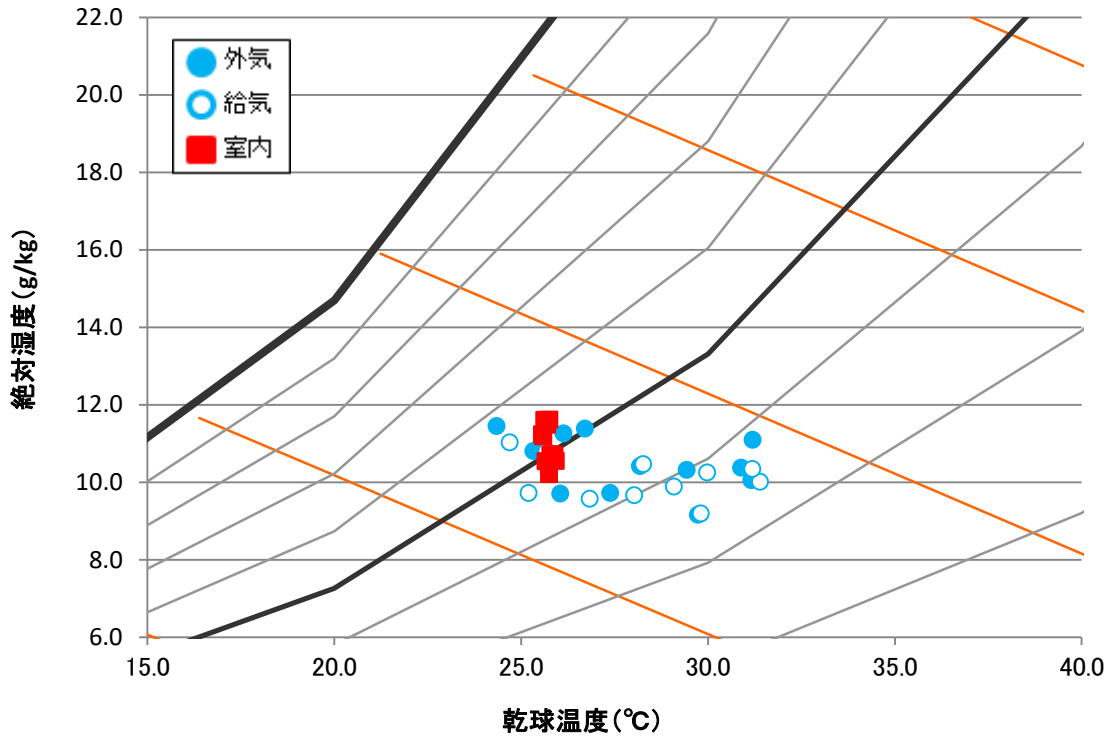


図 28 6/19 空気状態

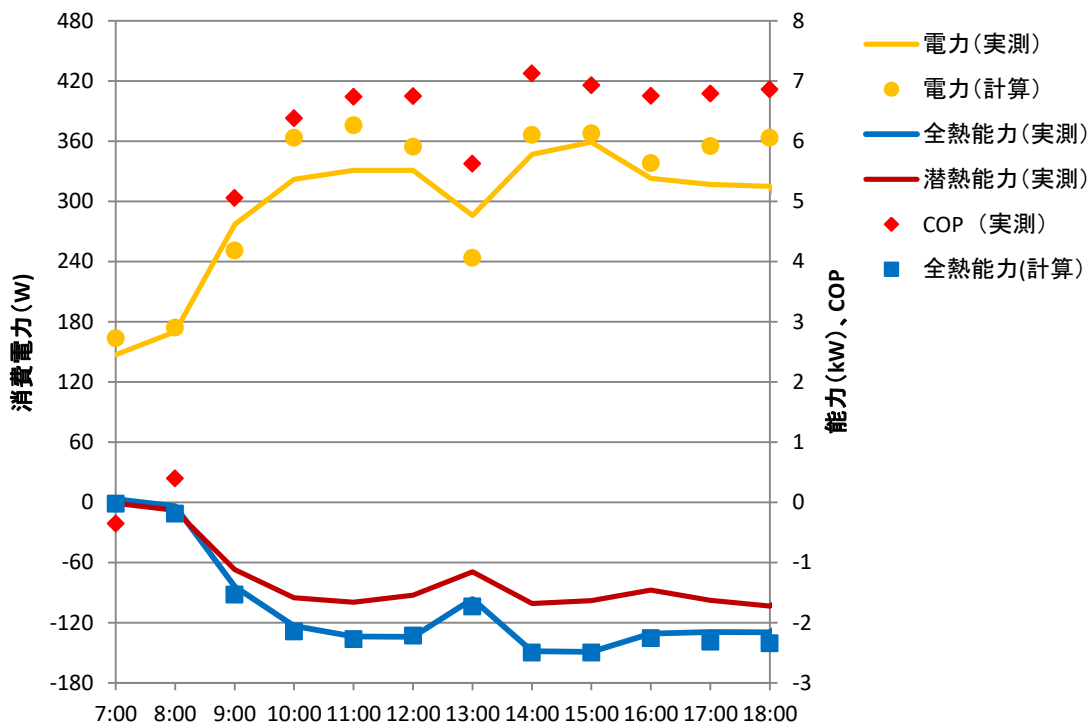


図 29 6/20 運転状態

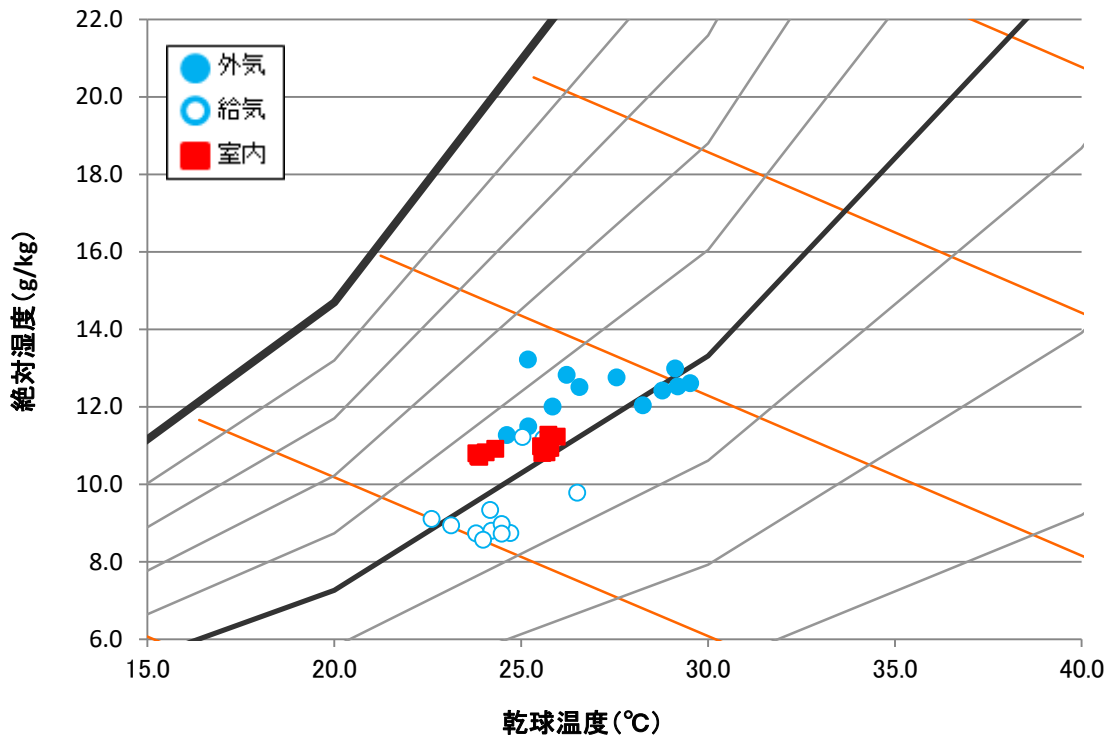


図 30 6/20 空気状態

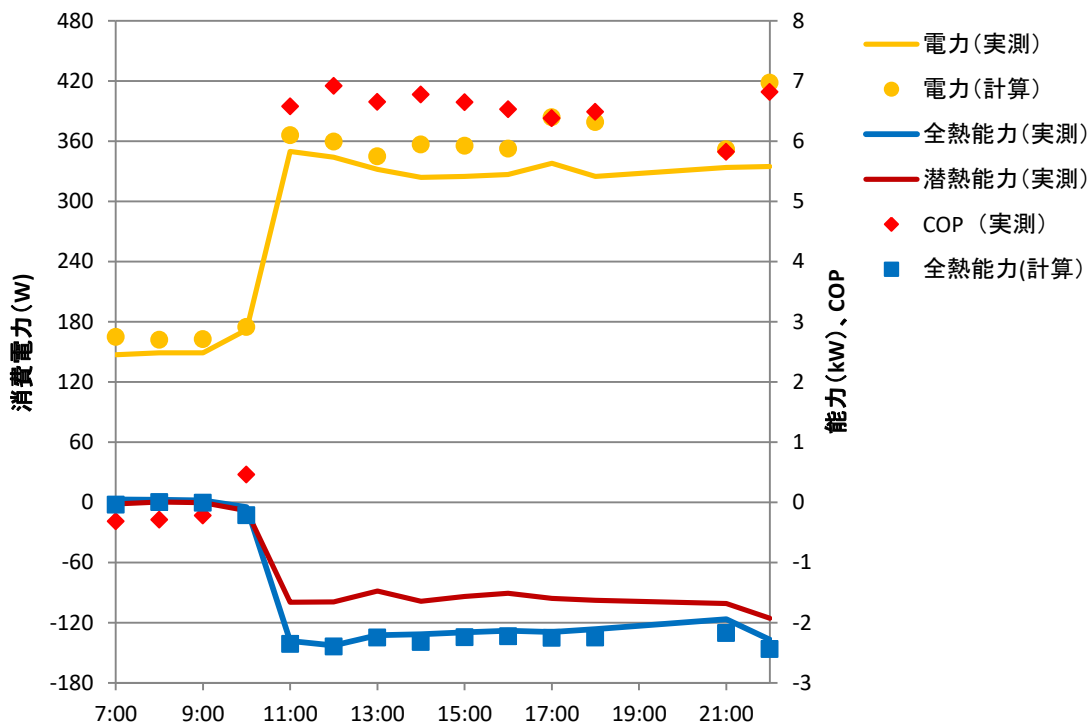


図 31 6/23 運転状態

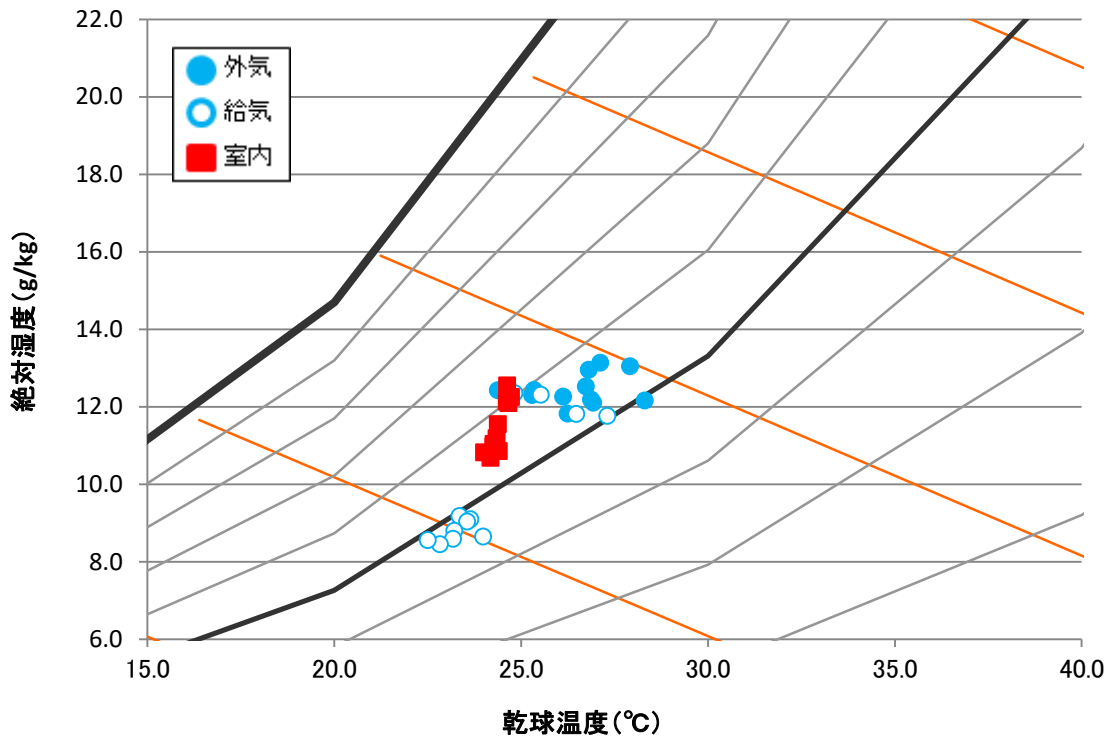


図 32 6/23 空気状態

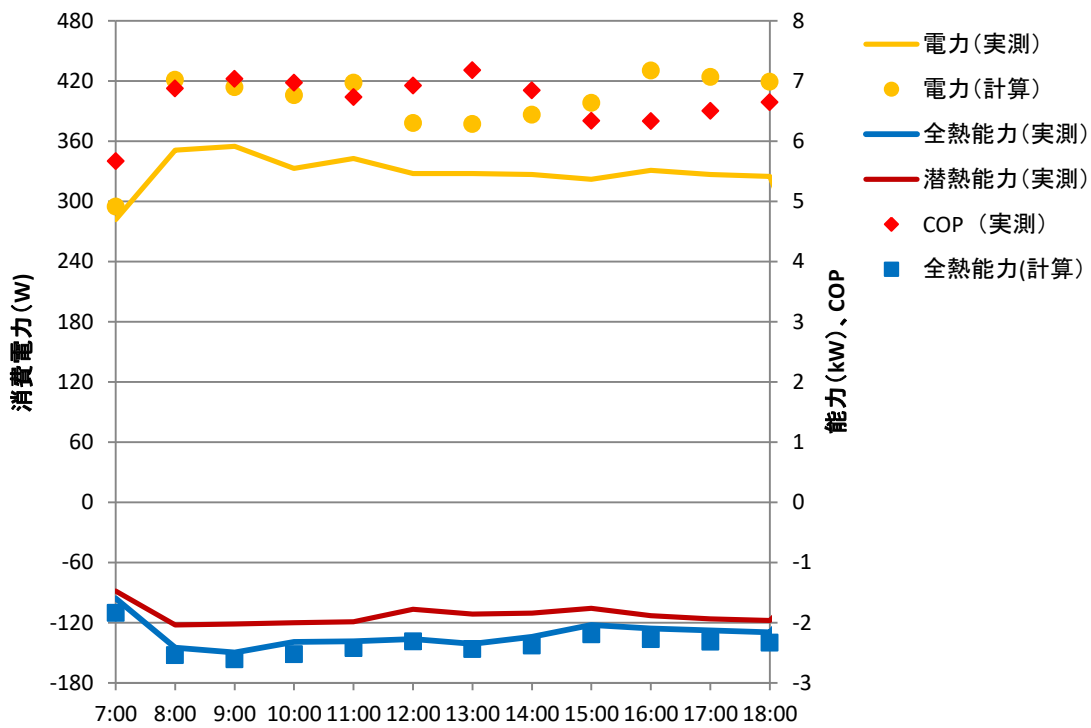


図 33 6/24 運転状態

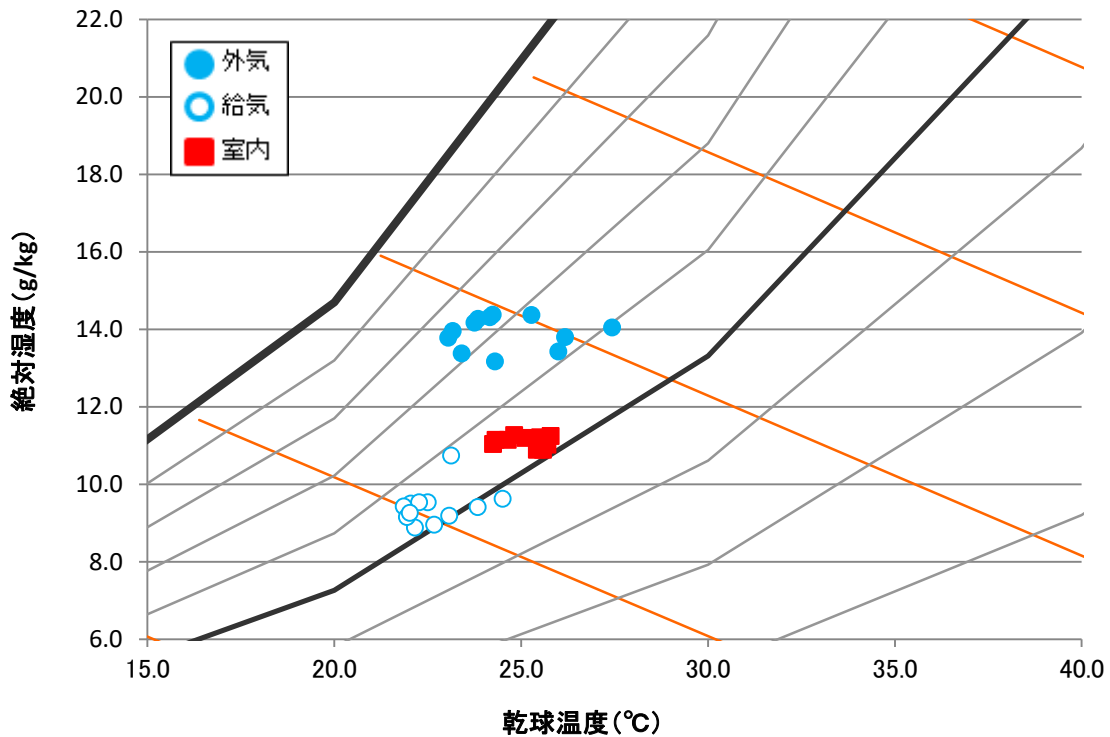


図 34 6/24 空気状態

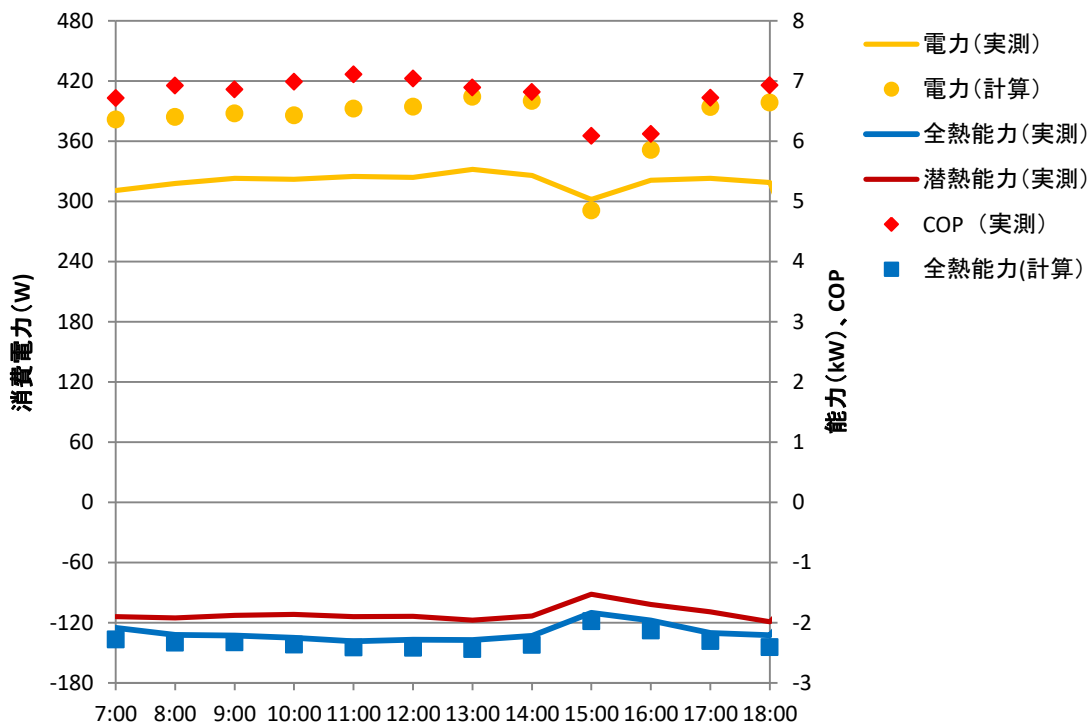


図 35 6/25 運転状態

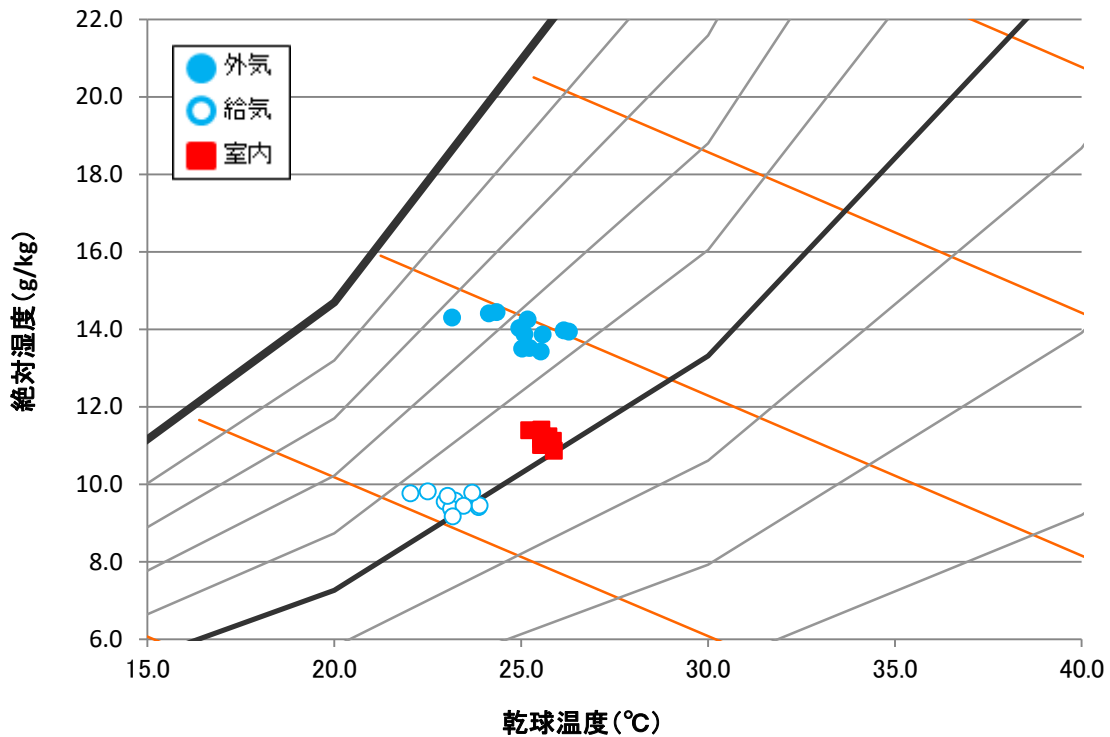


図 36 6/25 空気状態

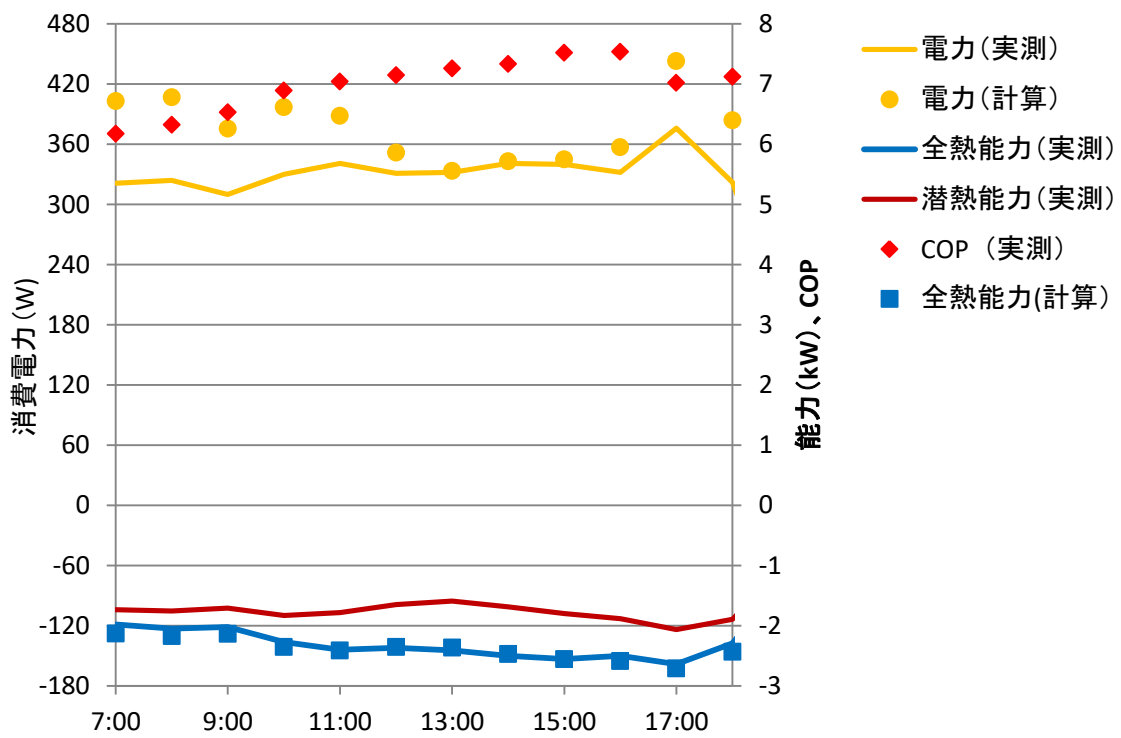


図 37 6/26 運転状態

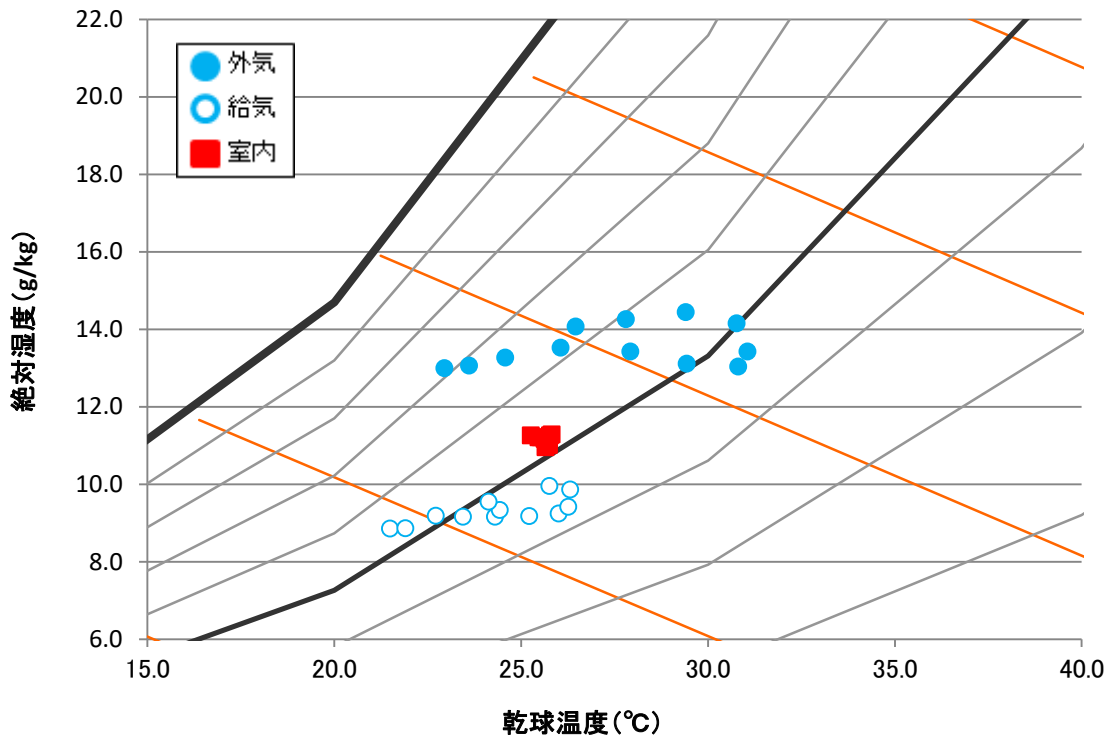


図 38 6/26 空気状態

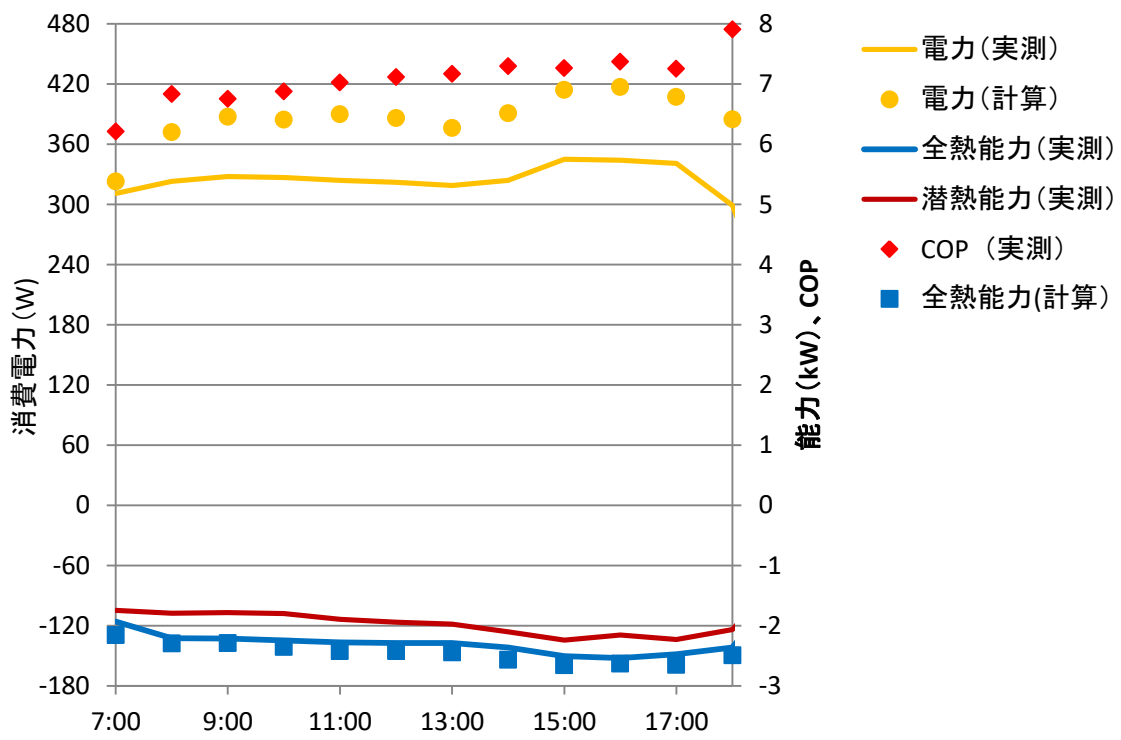


図 39 6/27 運転状態

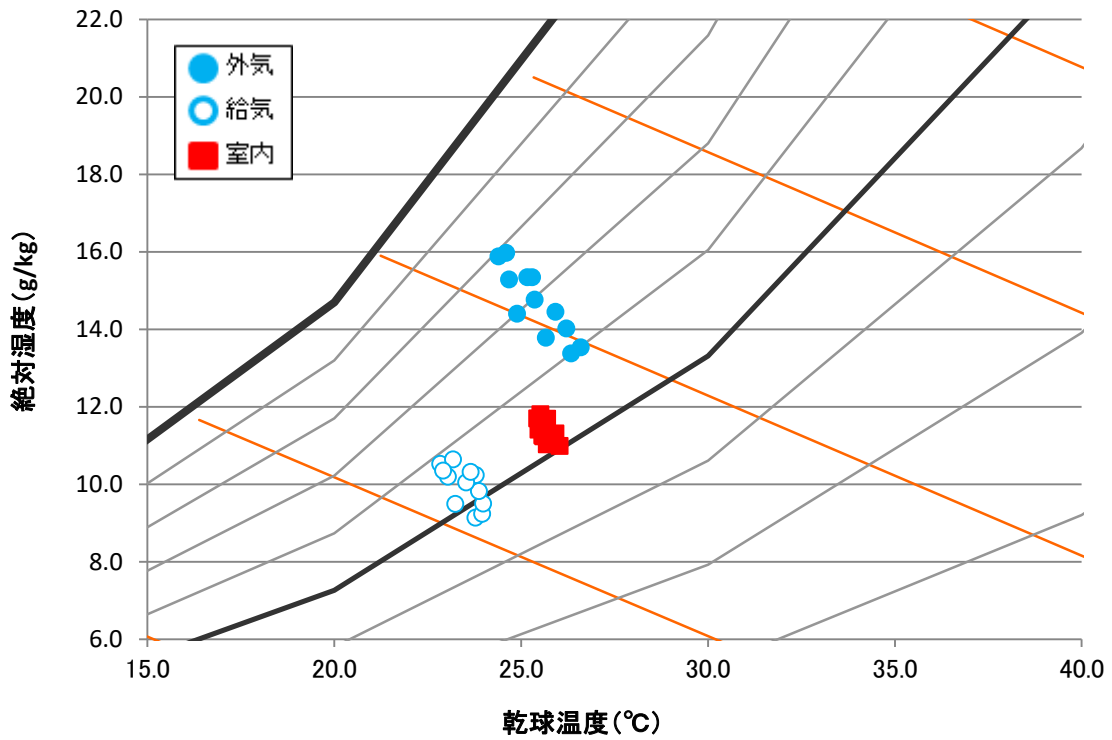


図 40 6/27 空気状態

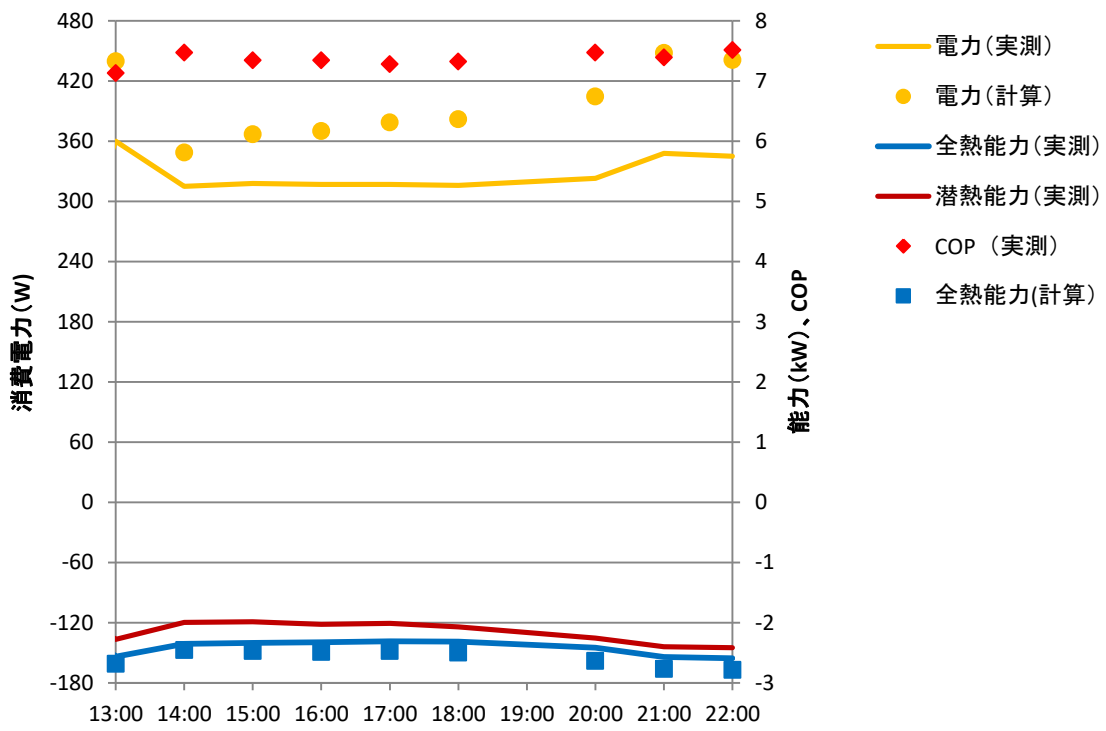


図 41 6/28 運転状態

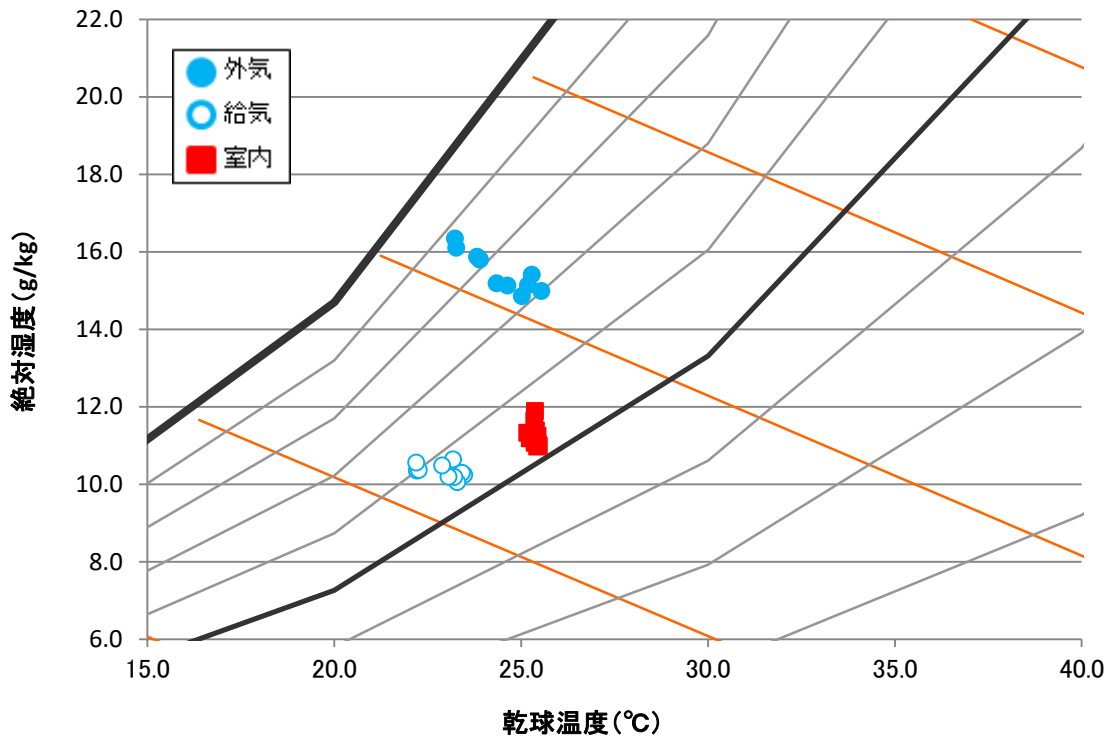


図 42 6/28 空気条件

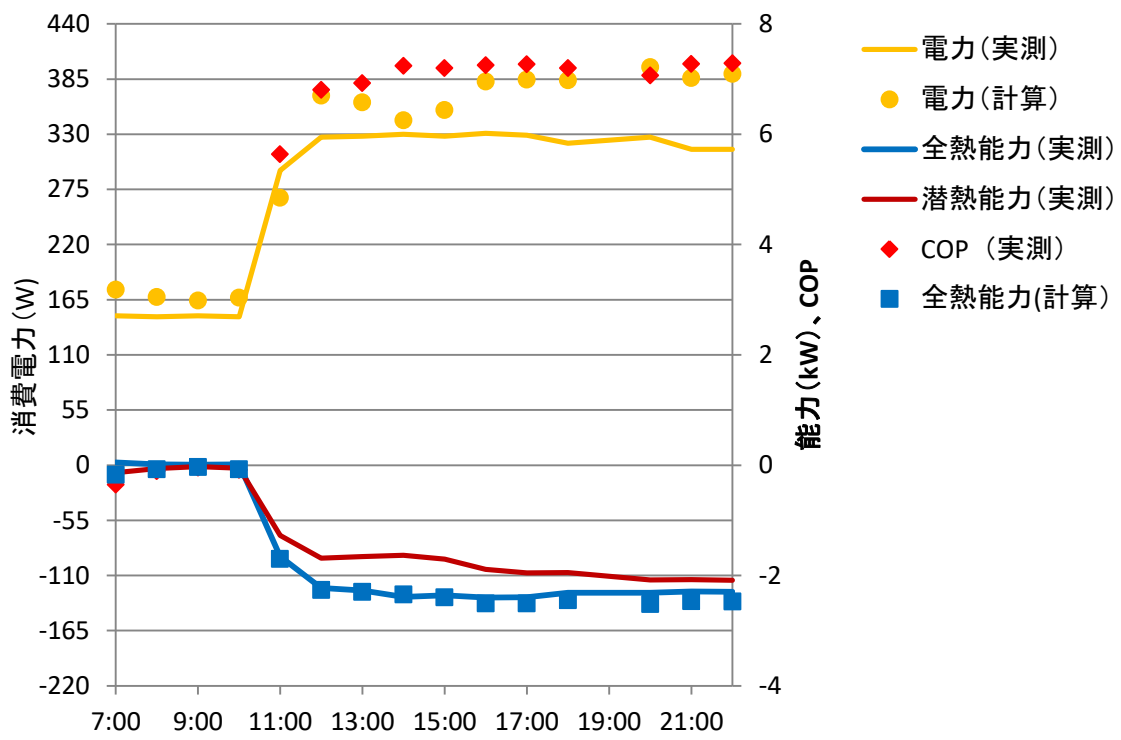


図 43 6/30 運転状態

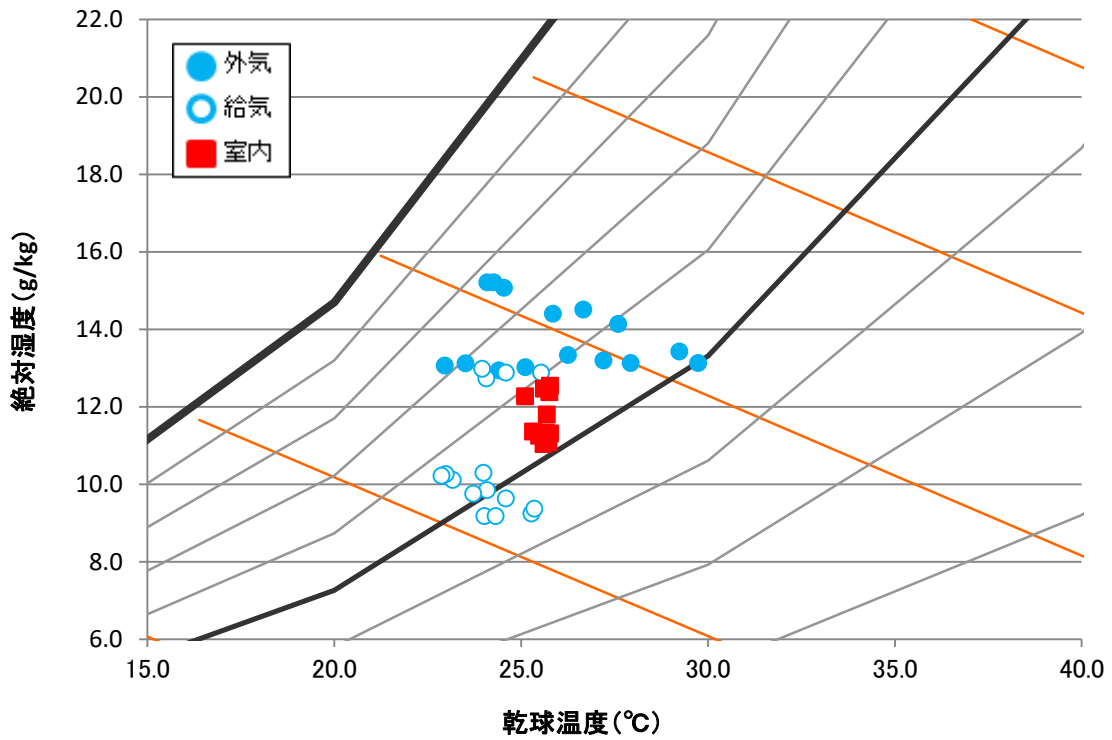


図 44 6/30 空気条件

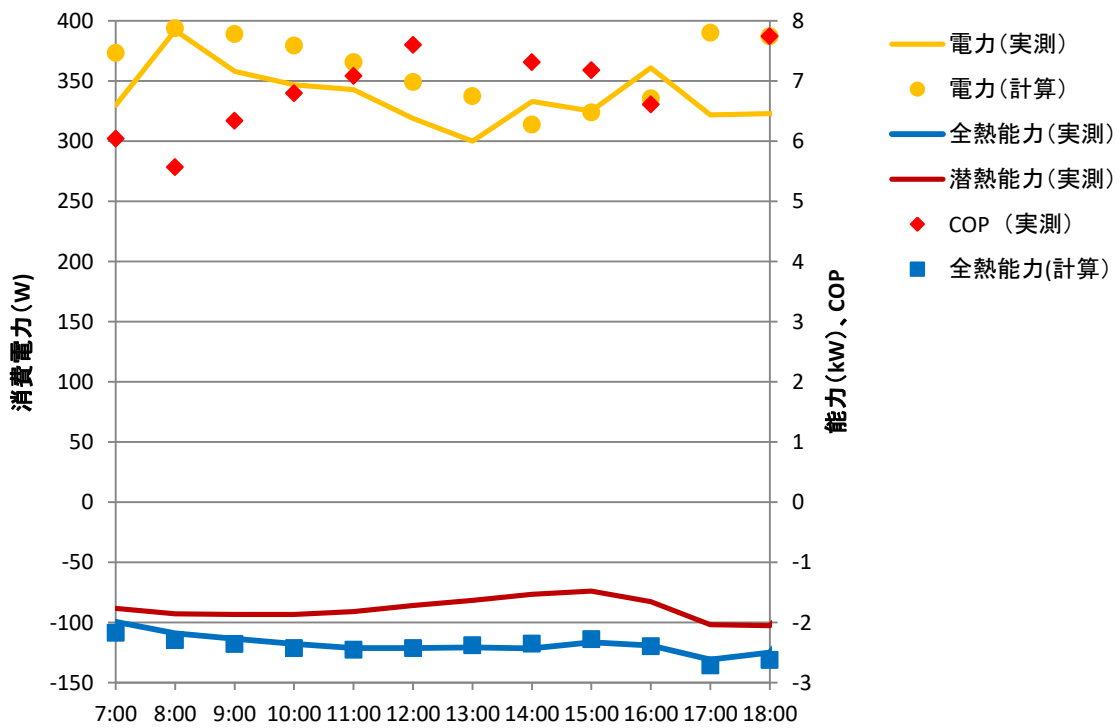


図 45 7/1 運転状態

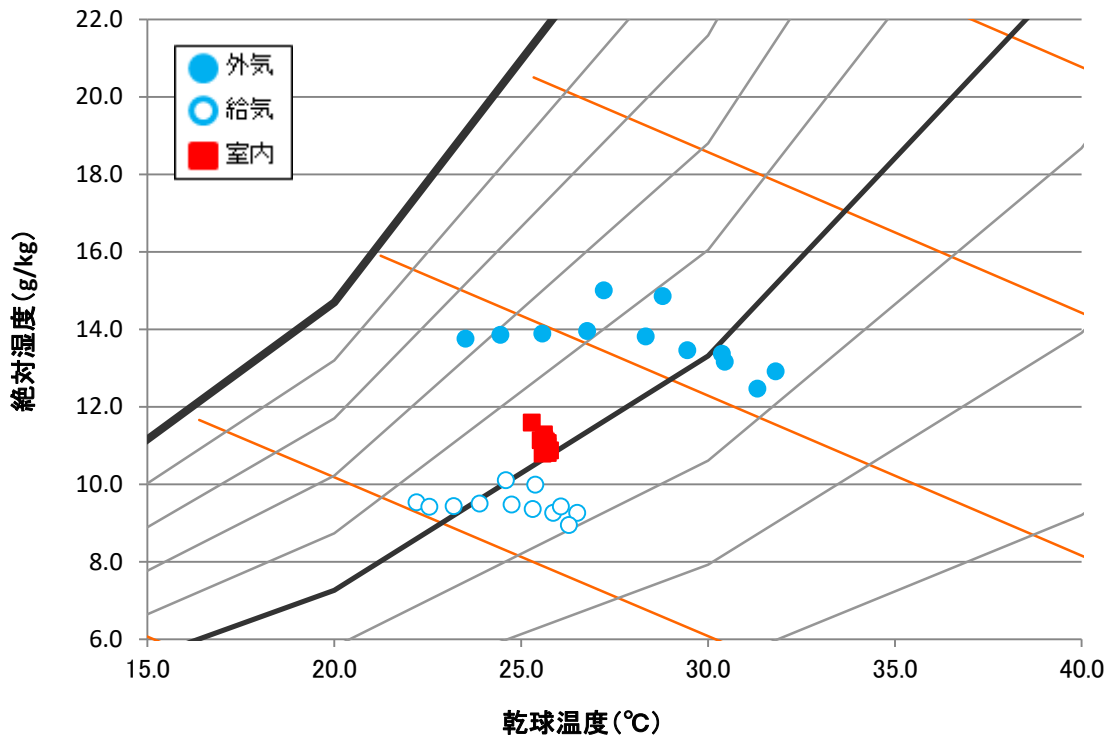


図 46 7/1 空気状態

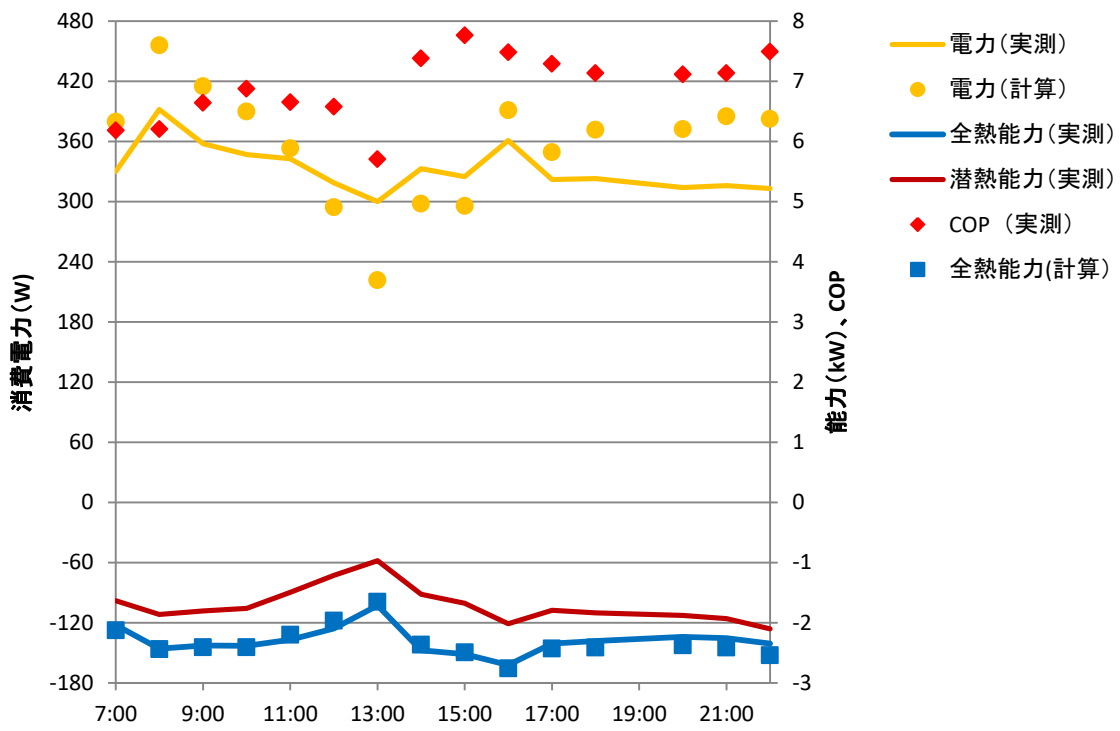


図 47 7/2 運転状態

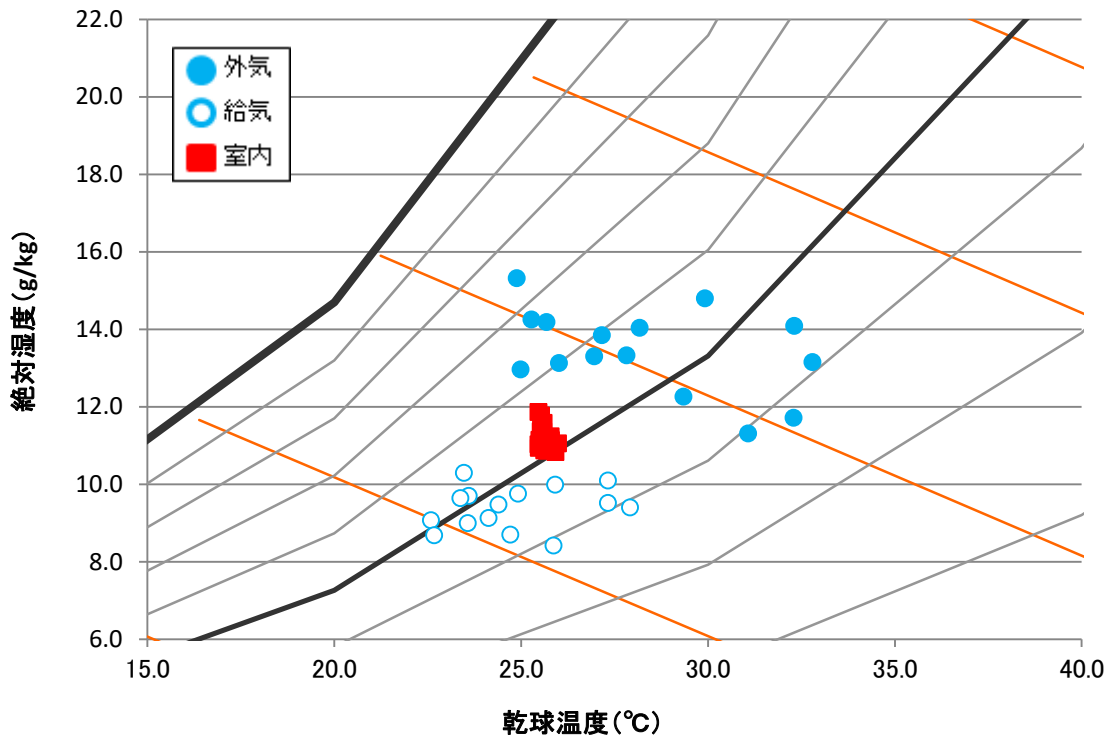


図 48 7/2 空気状態

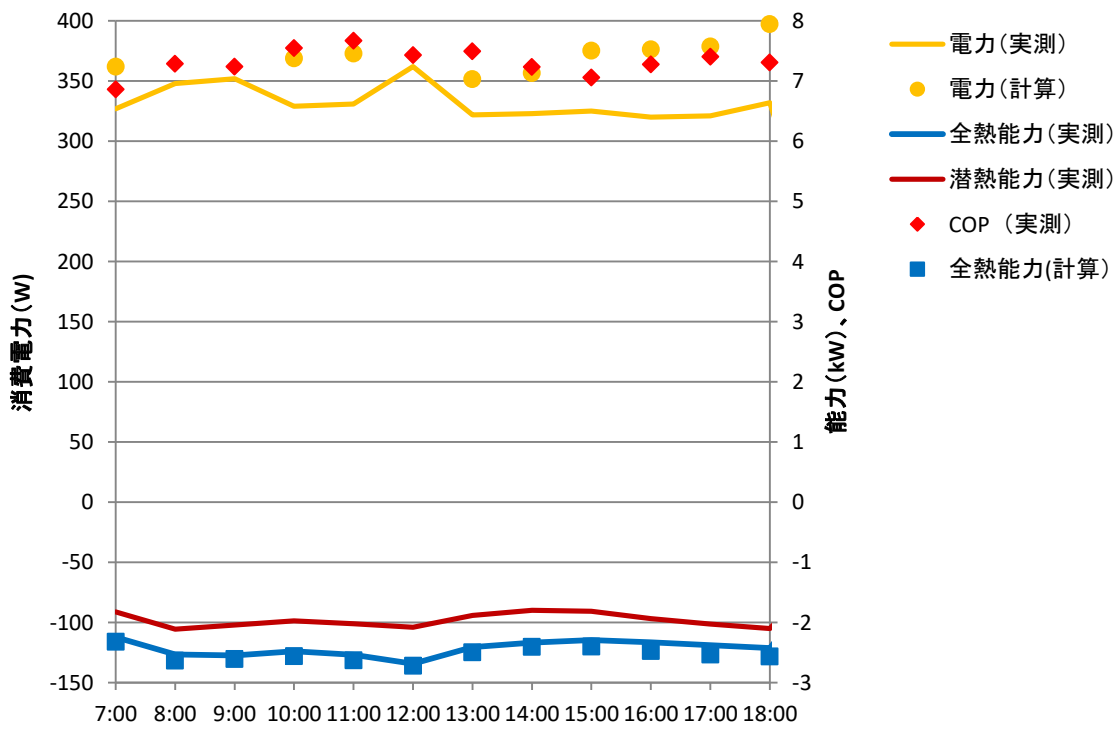


図 49 7/3 運転状態

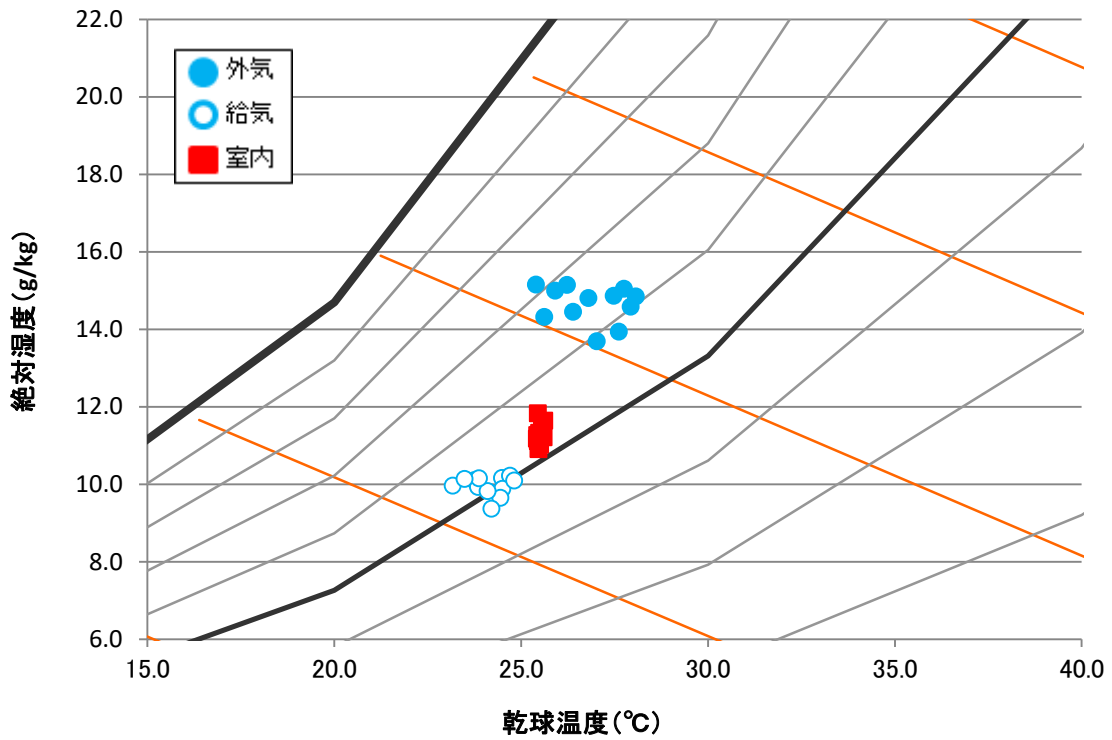


図 50 7/3 空気状態

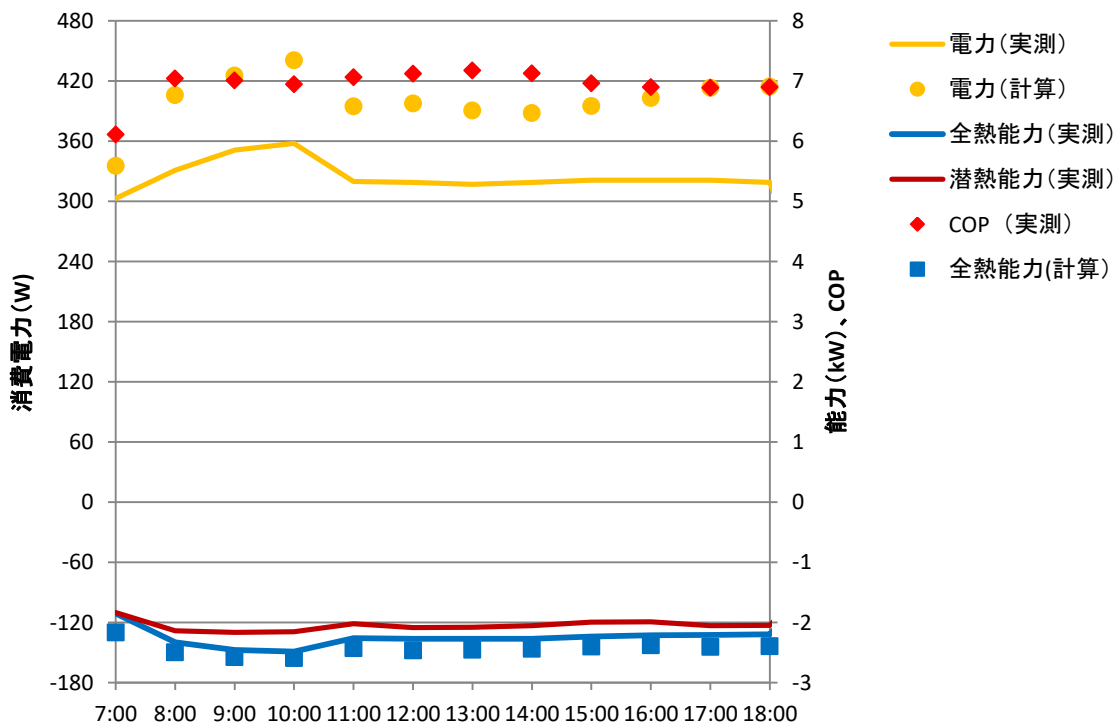


図 51 7/4 運転状態

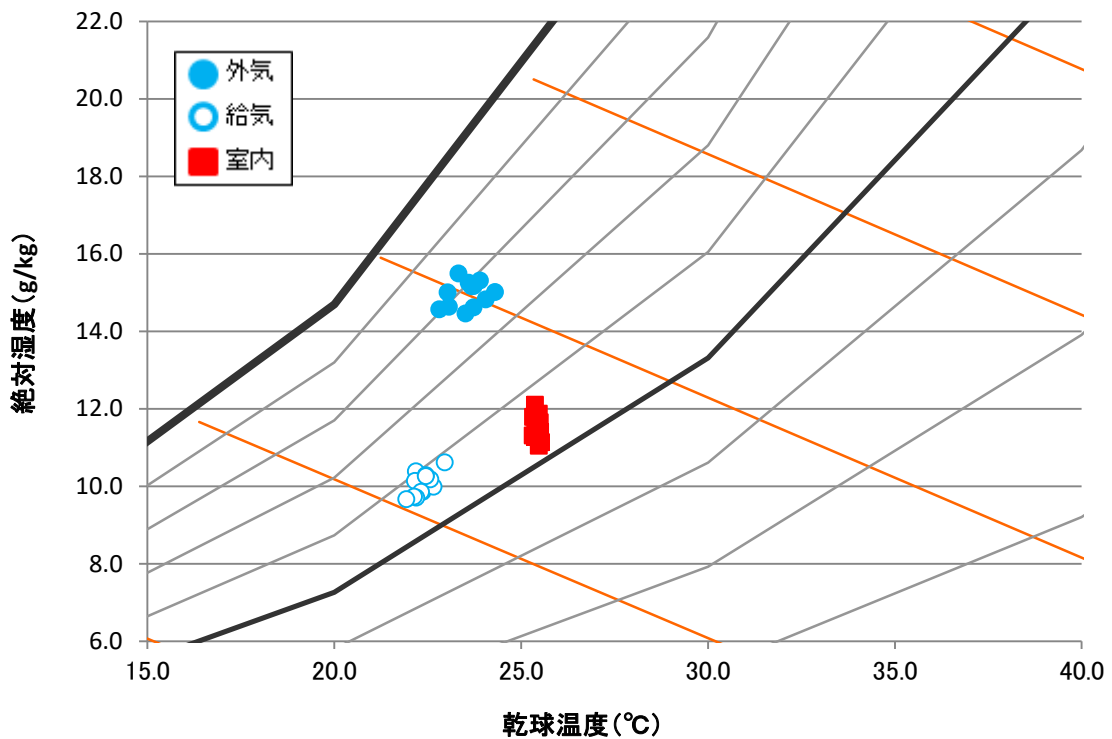


図 52 7/4 空気状態

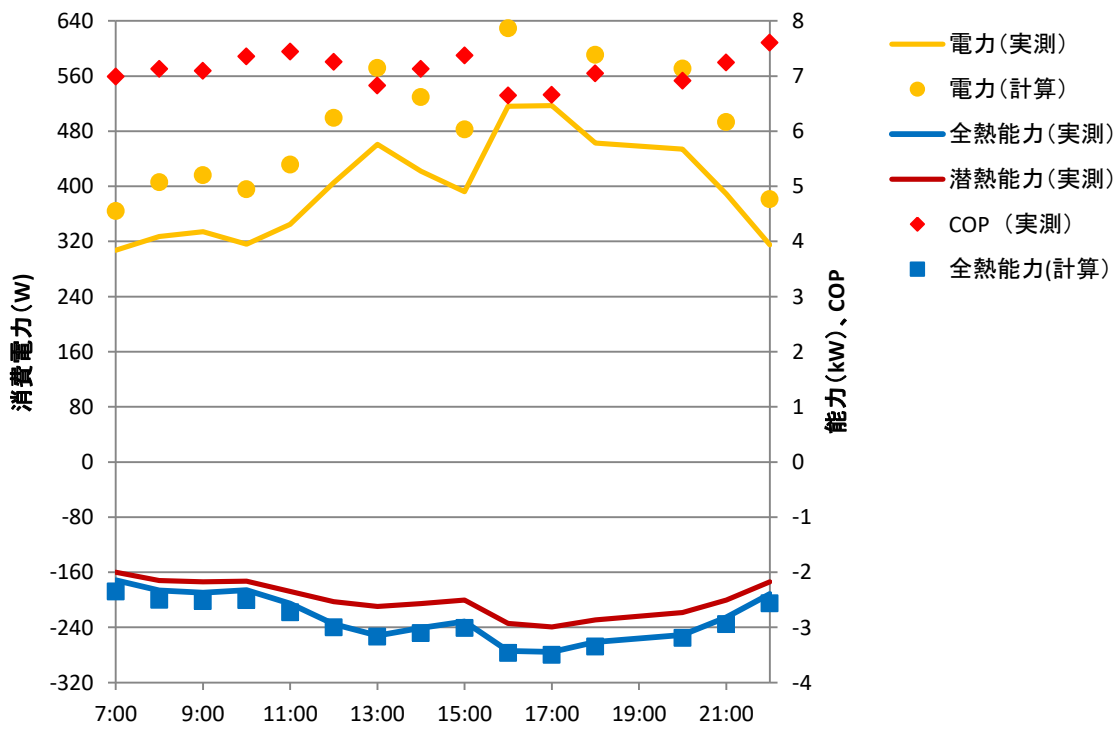


図 53 7/7 運転状態

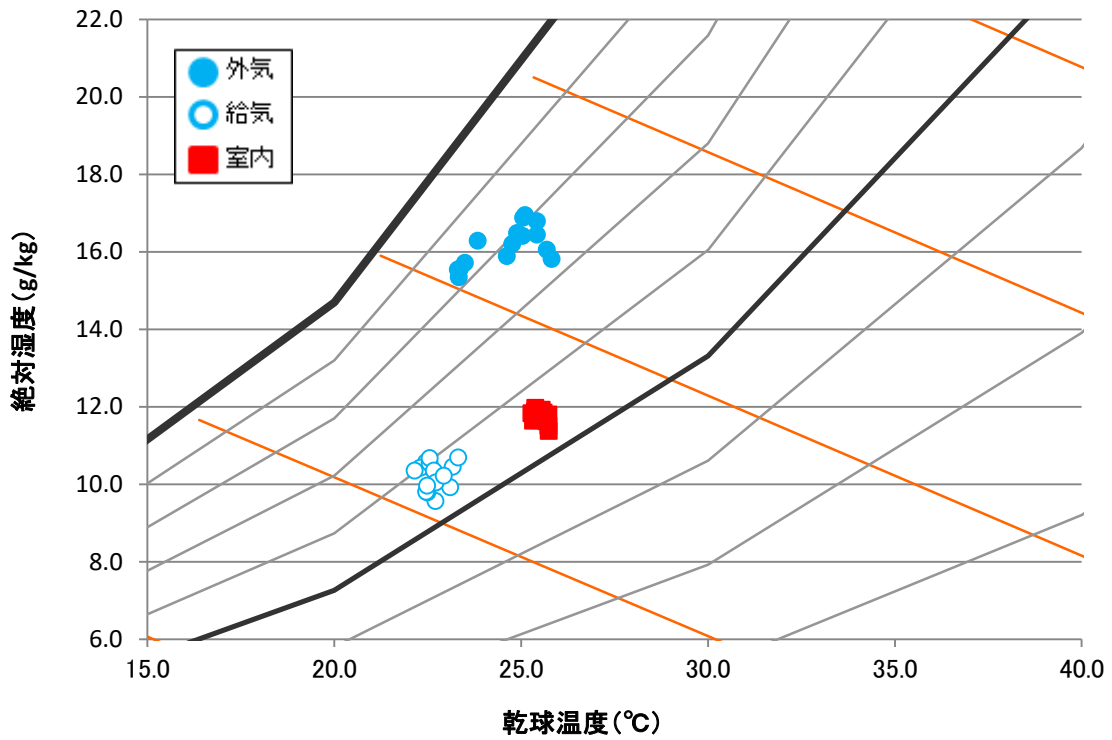


図 54 7/7 空気状態

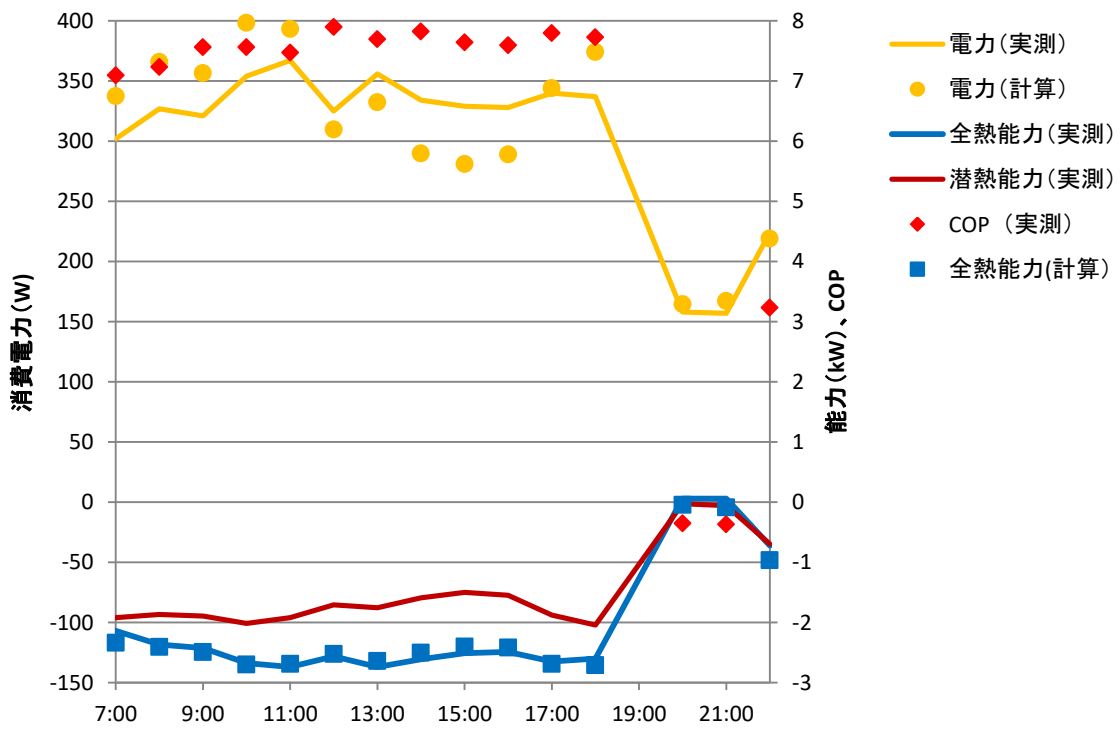


図 55 7/8 運転状態

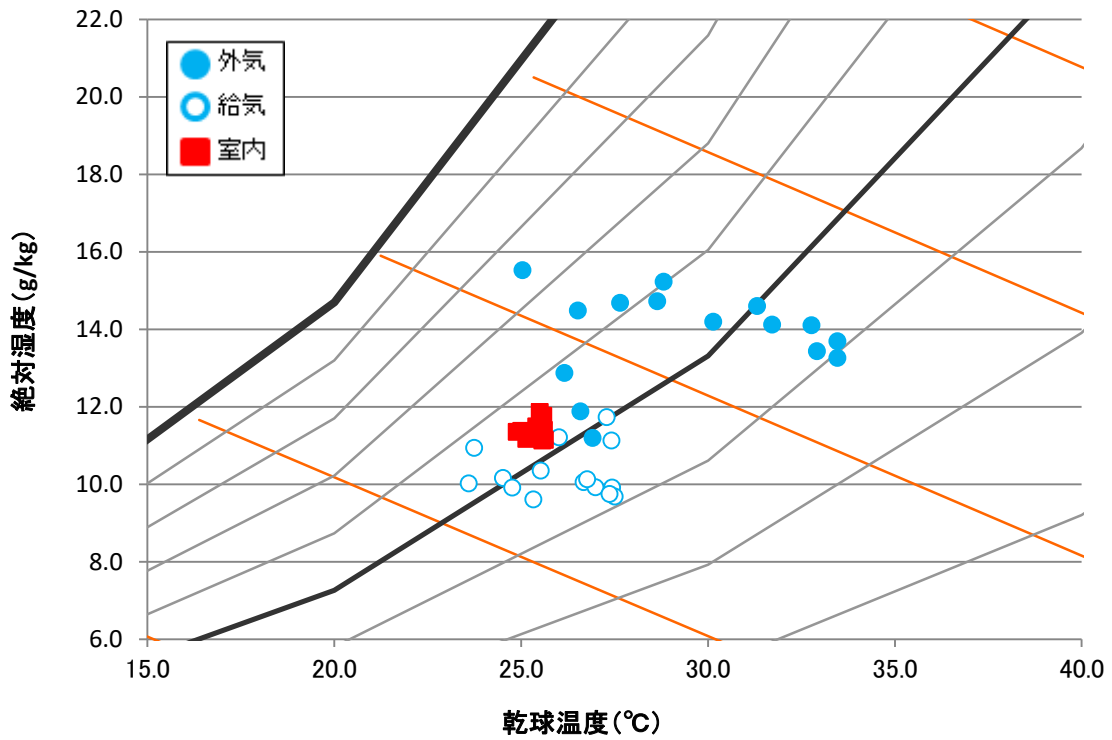


図 56 7/8 空気状態

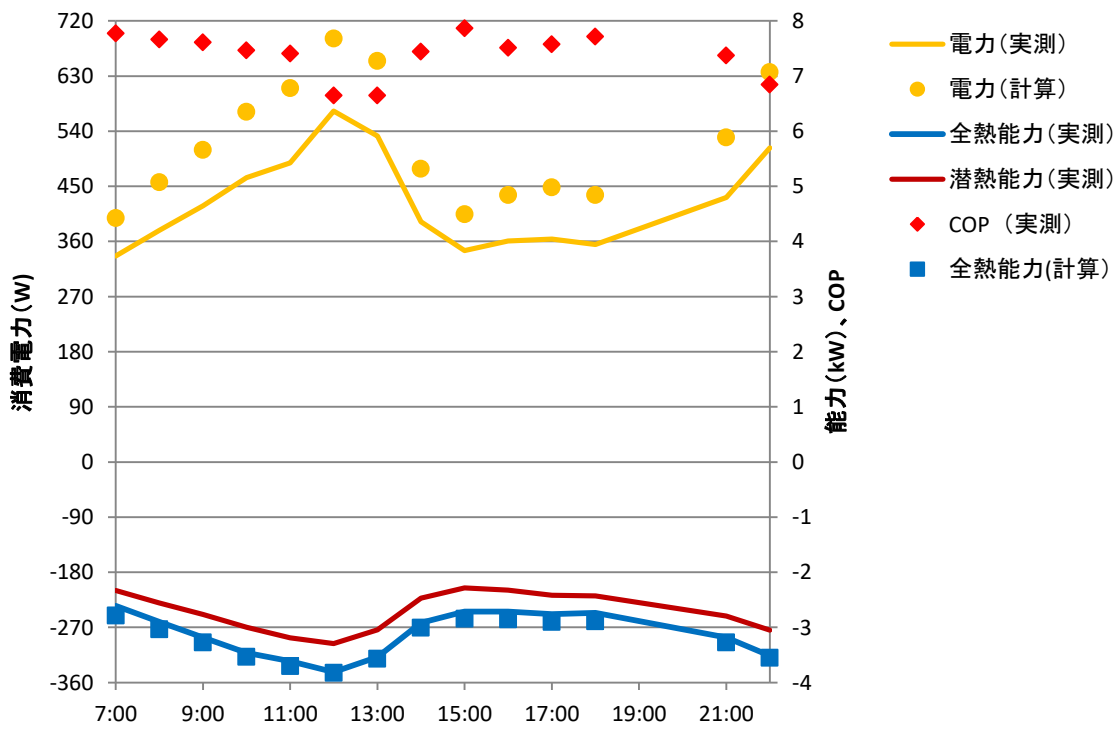


図 57 7/9 運転状態

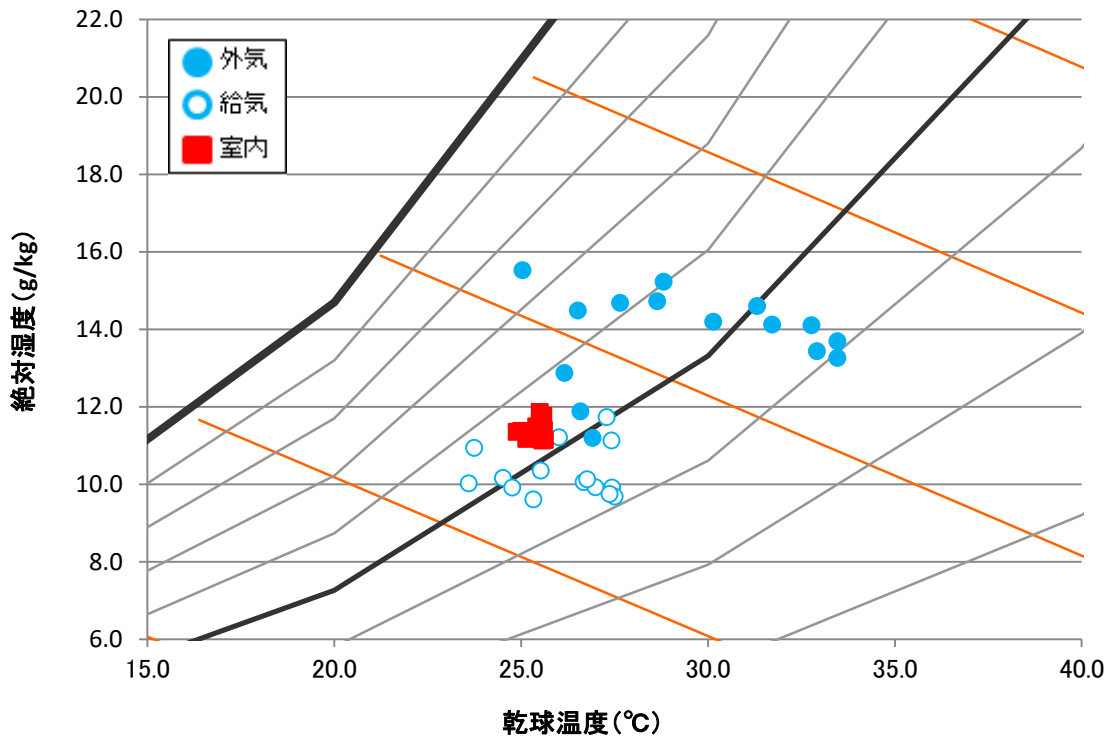


図 58 7/9 空気状態

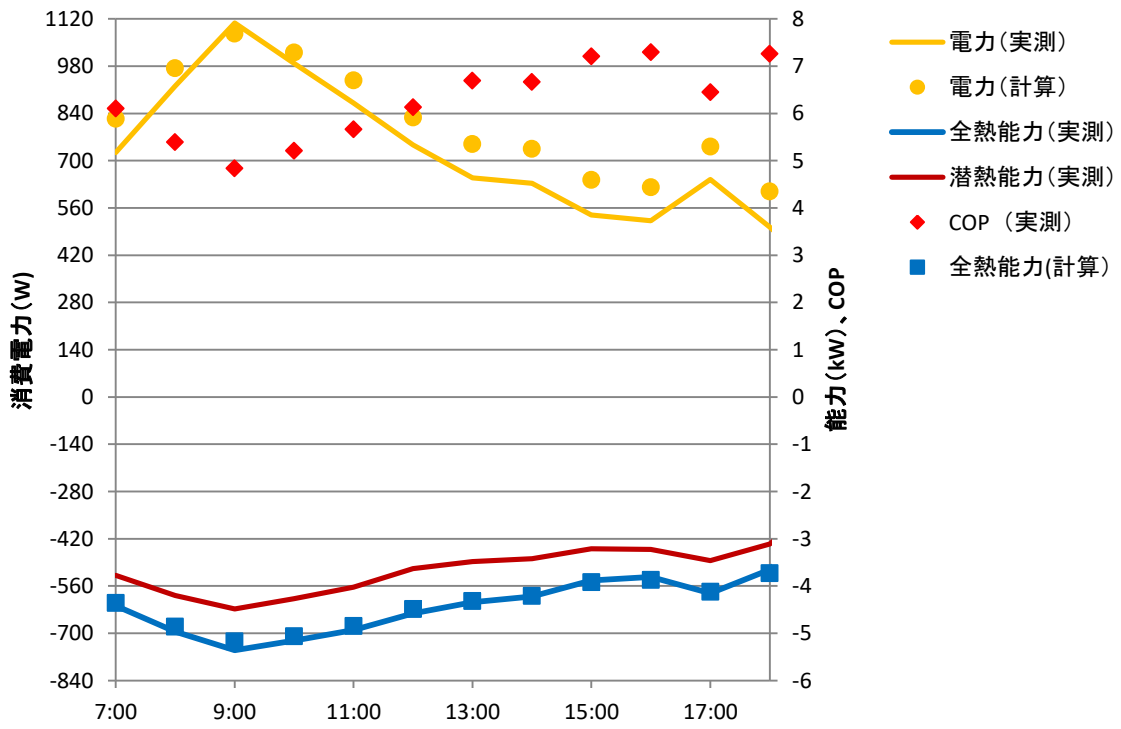


図 59 7/10 運転状態

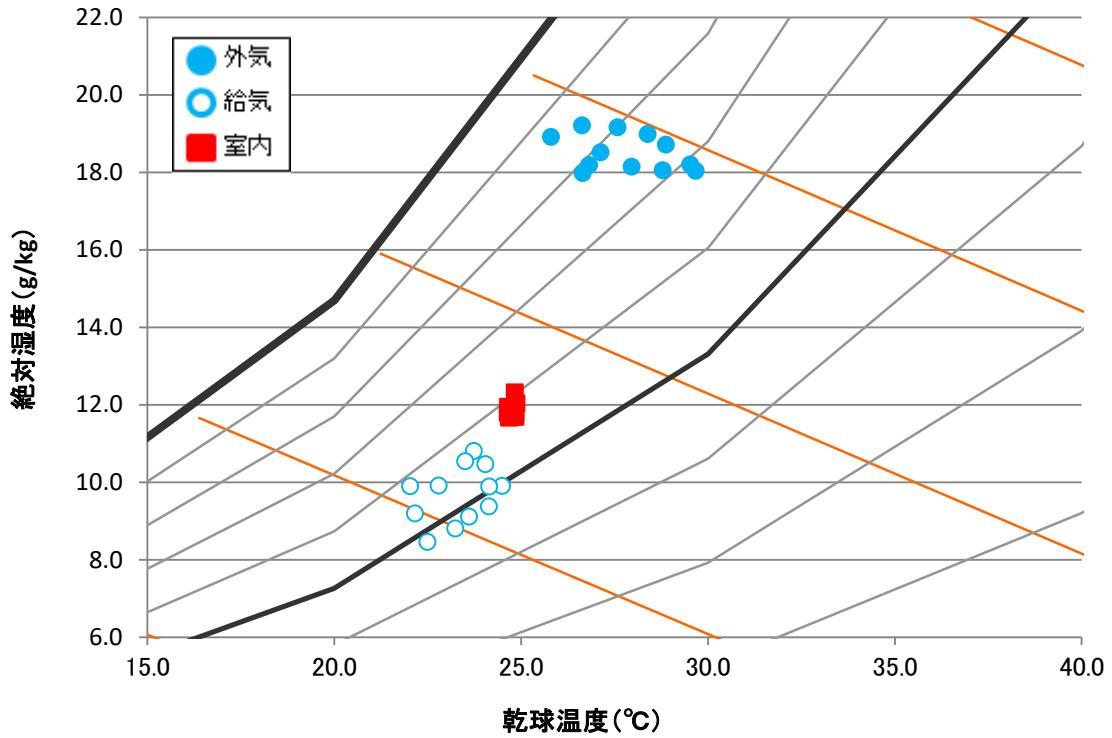


図 60 7/10 空気状態

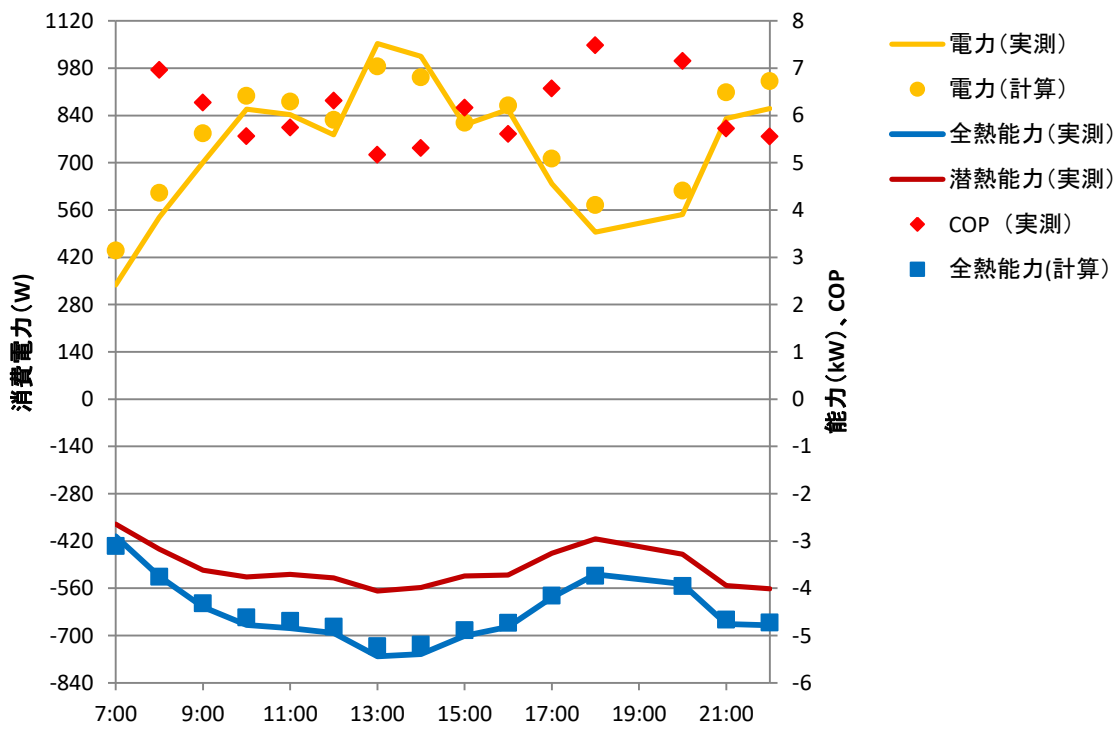


図 61 7/11 運転状態

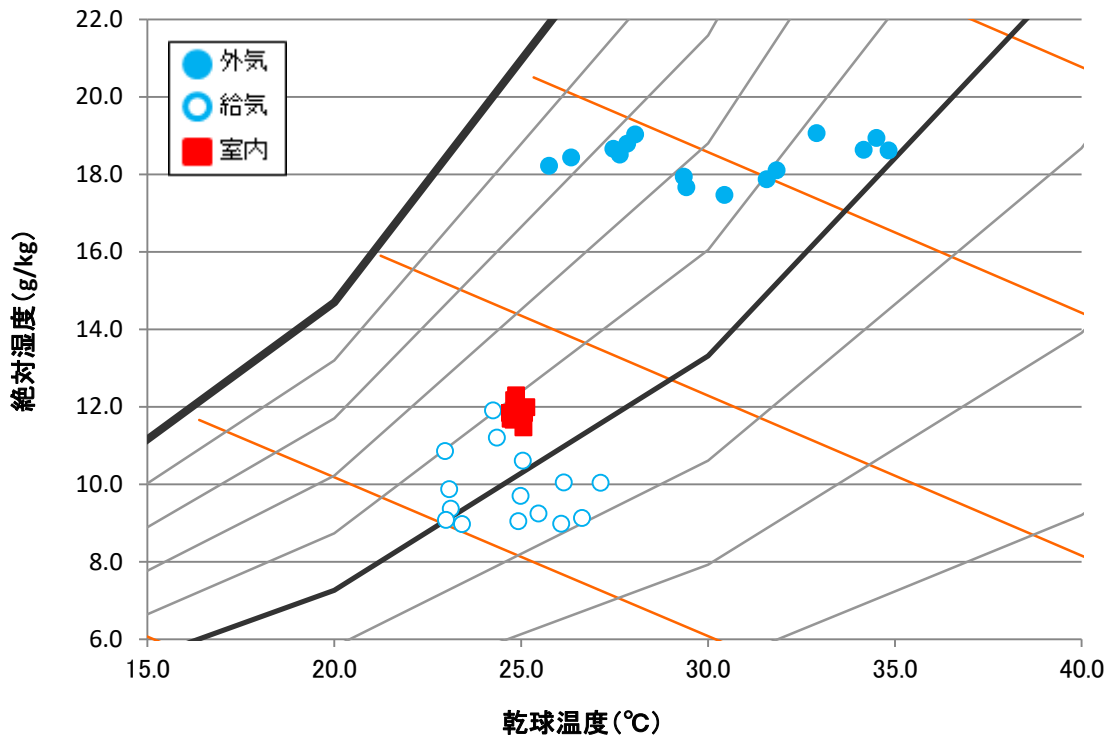


図 62 7/11 空気状態

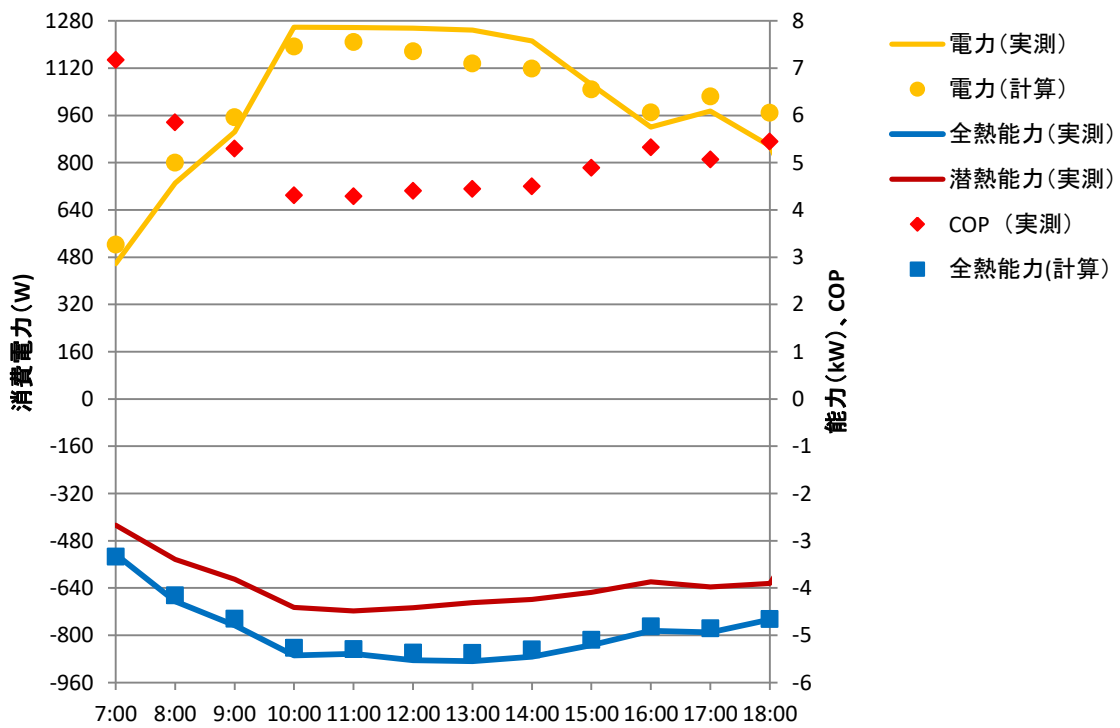


図 63 7/14 運転状態

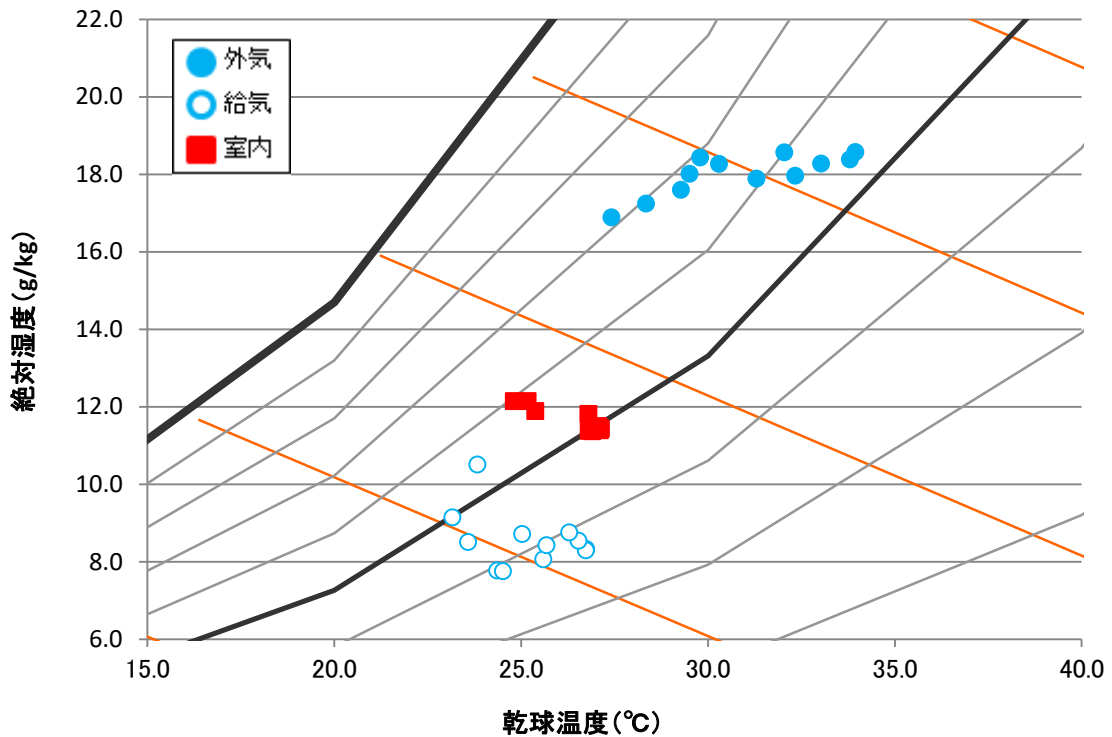


図 64 7/14 空気状態

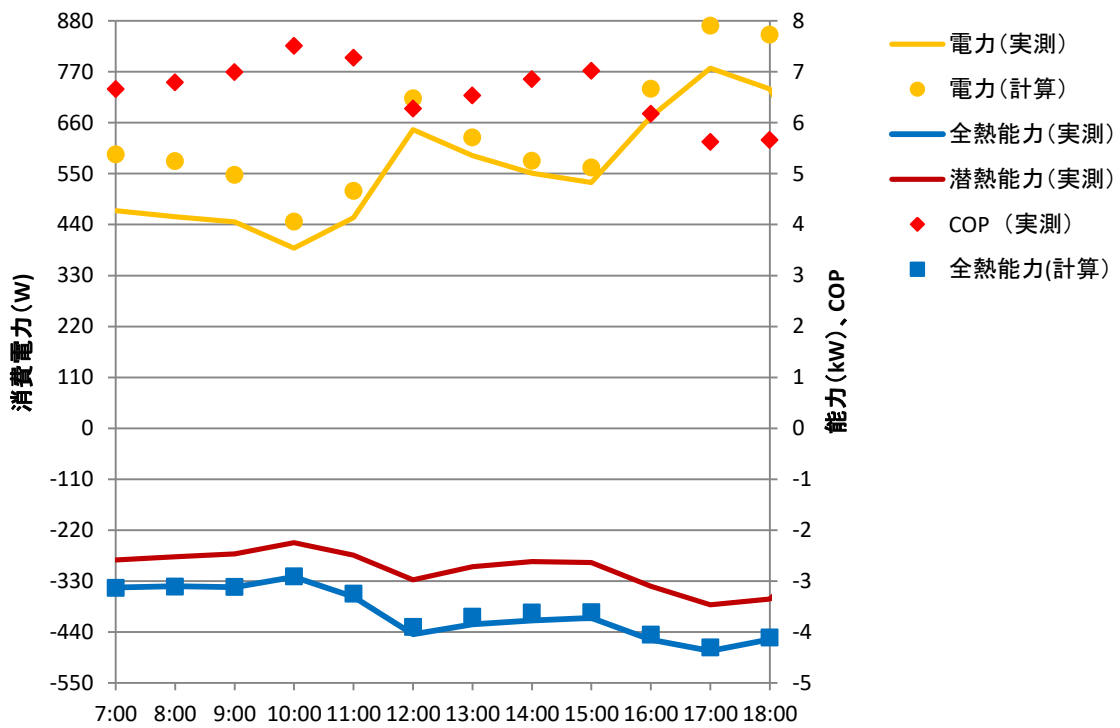


図 65 7/15 運転状態

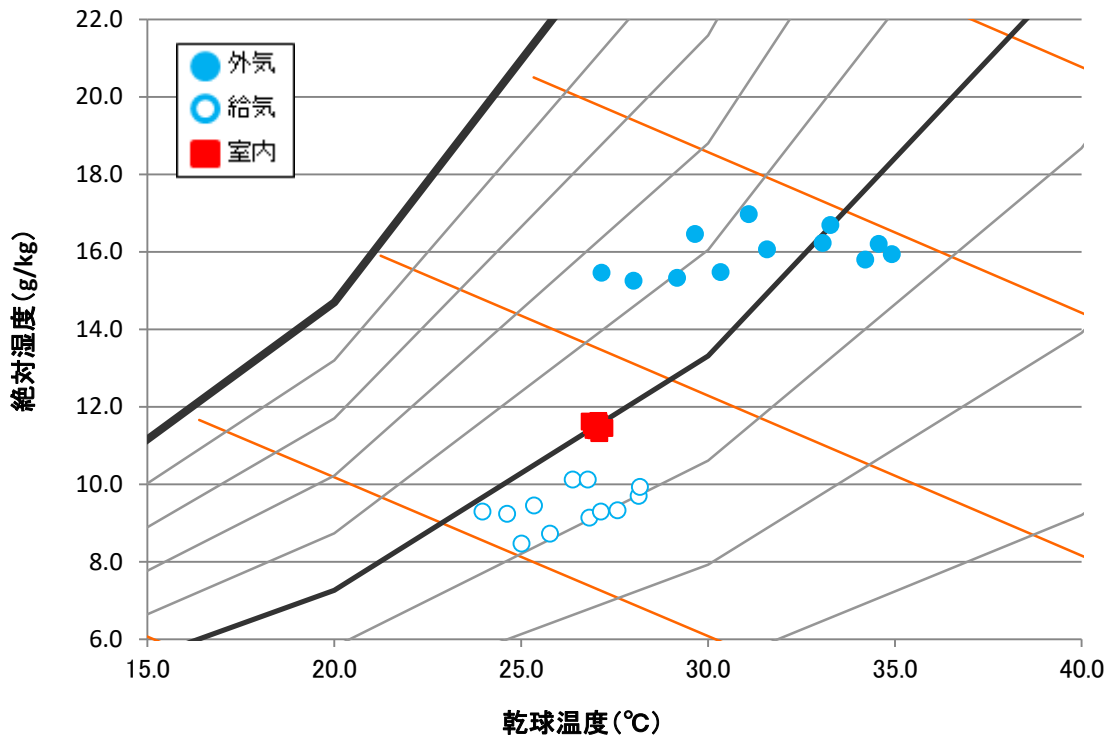


図 66 7/15 空気状態

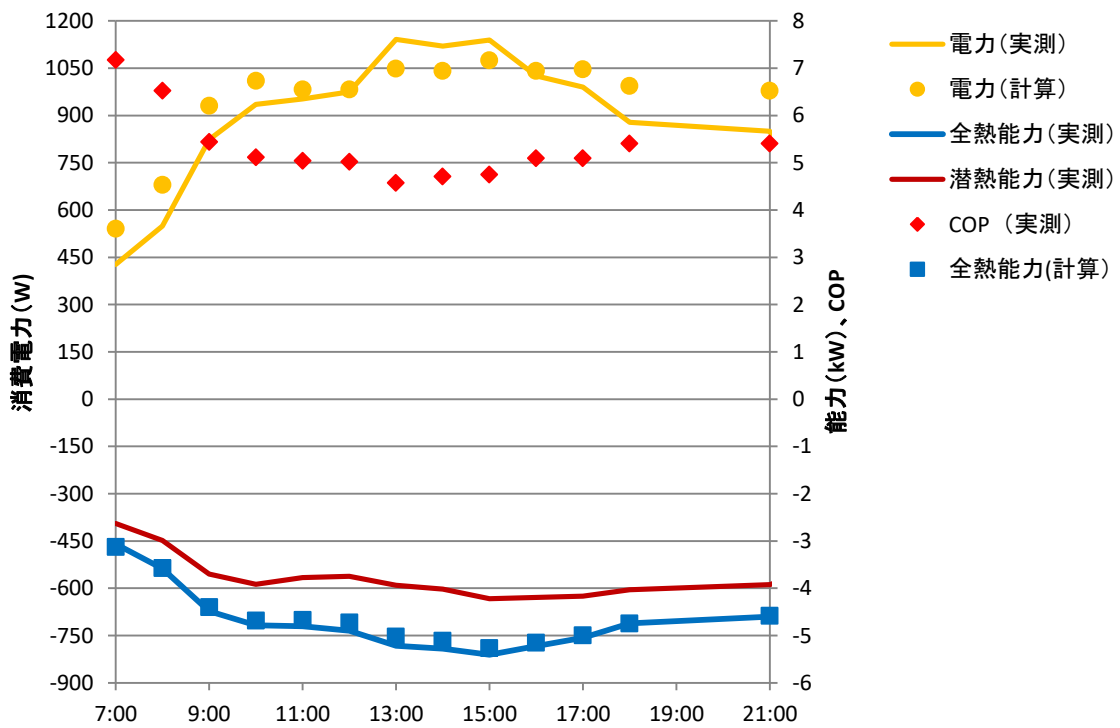


図 67 7/16 運転状態

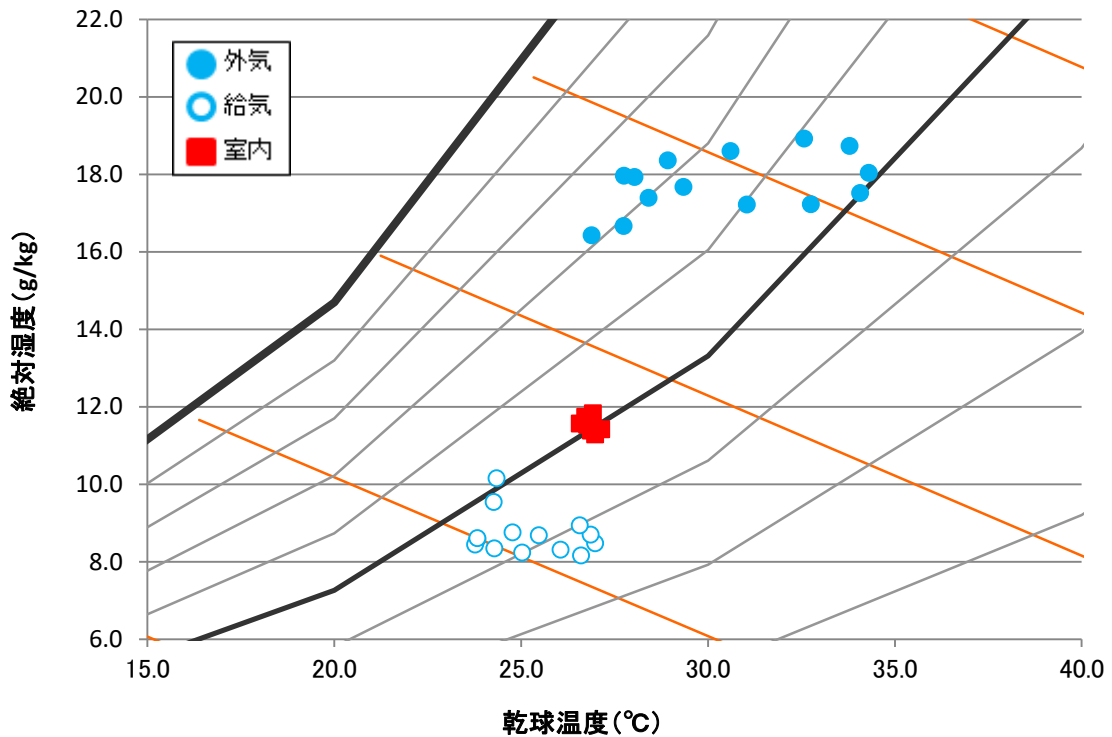


図 68 7/16 空気状態

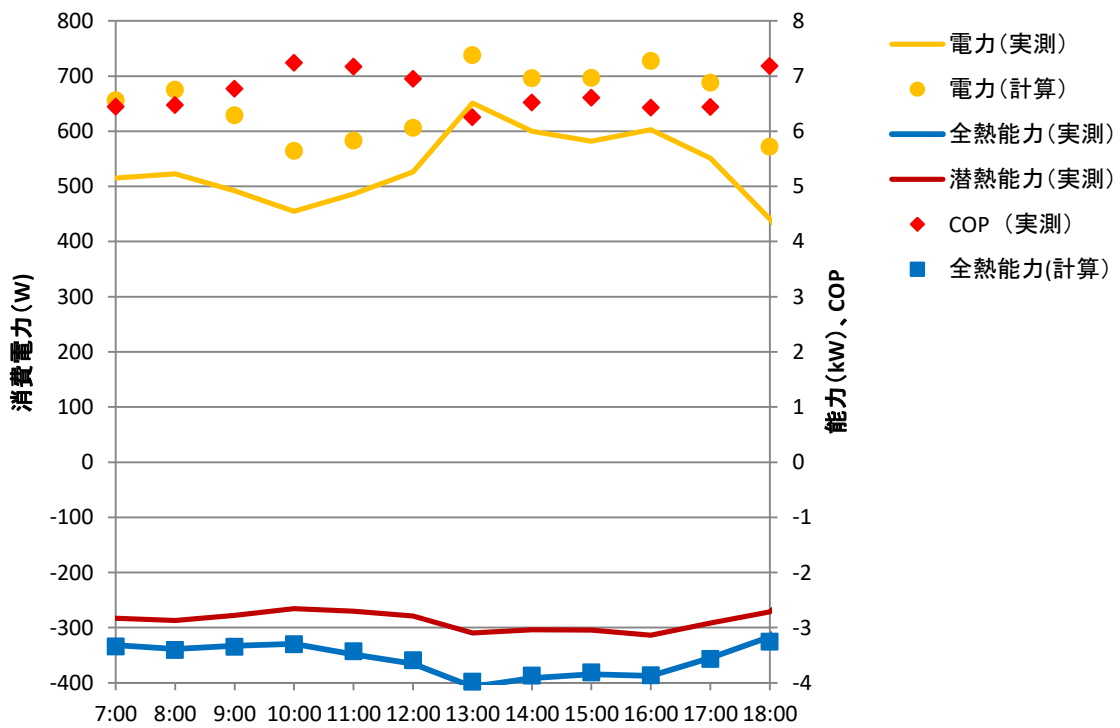


図 69 7/17 運転状態

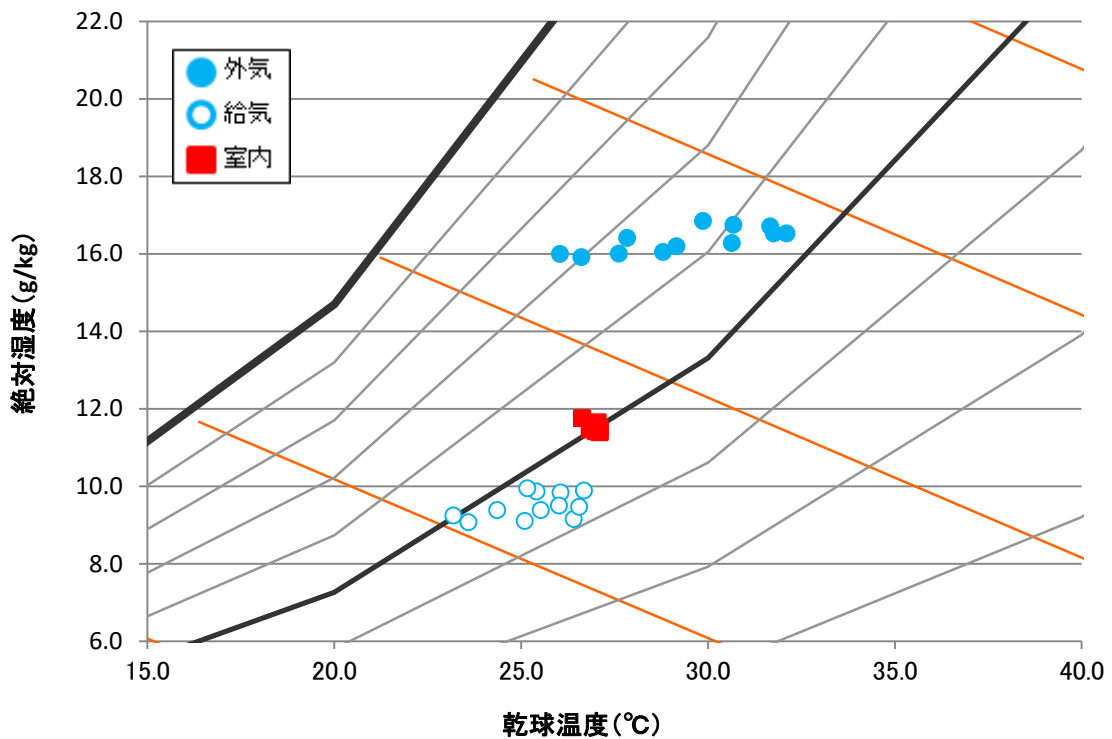


図 70 7/17 空気状態

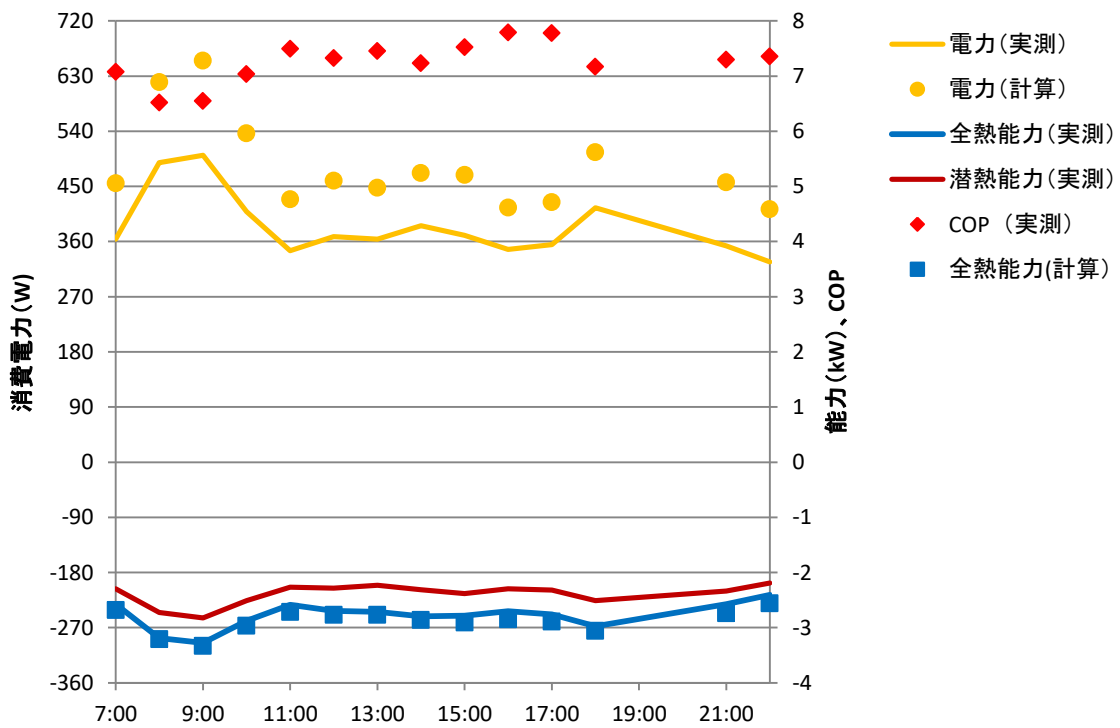


図 71 7/18 運転状態

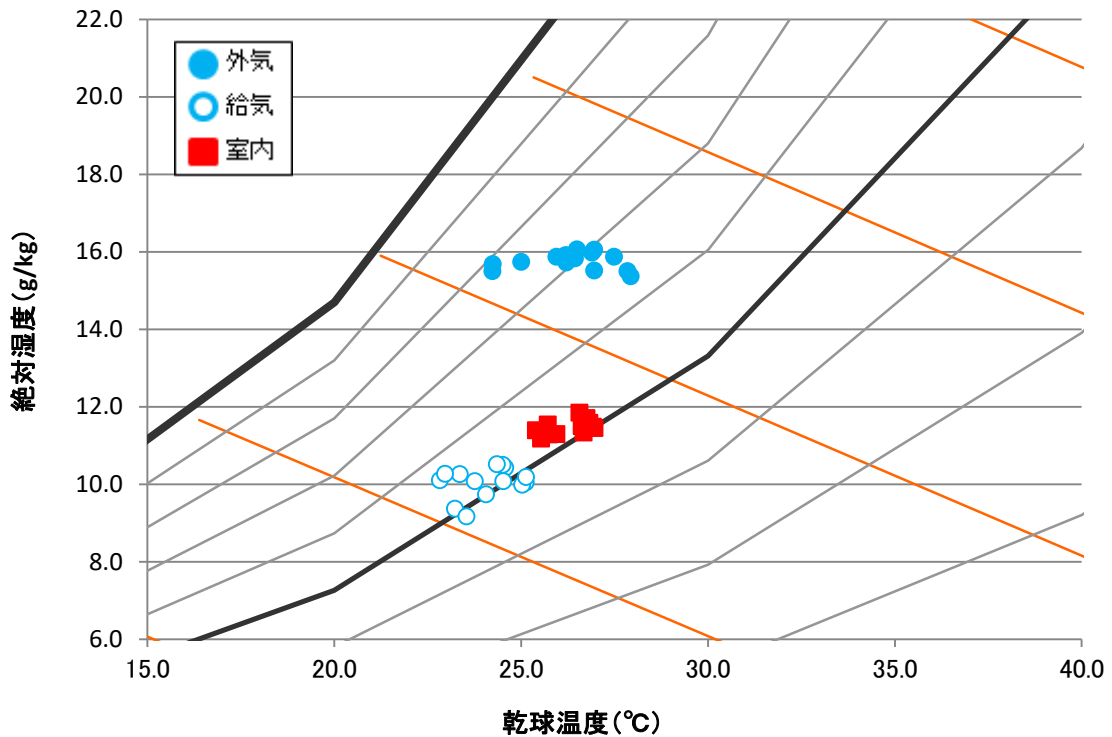


図 72 7/18 空気状態

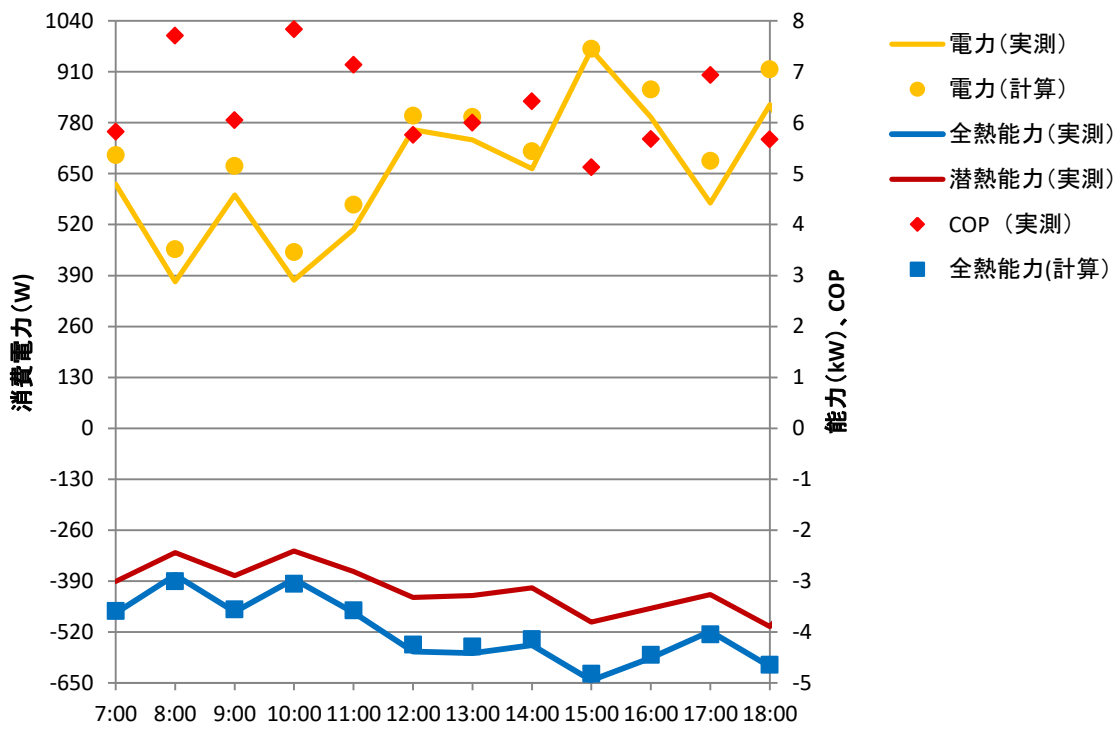


図 73 7/22 運転状態

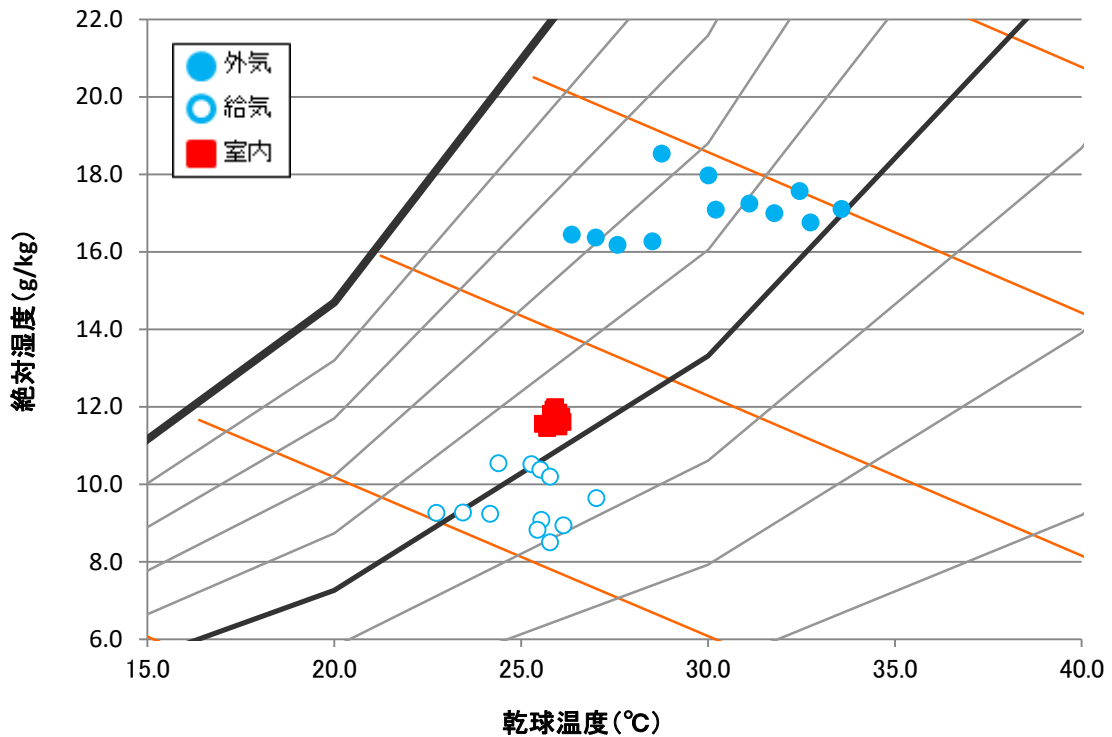


図 74 7/22 空気状態

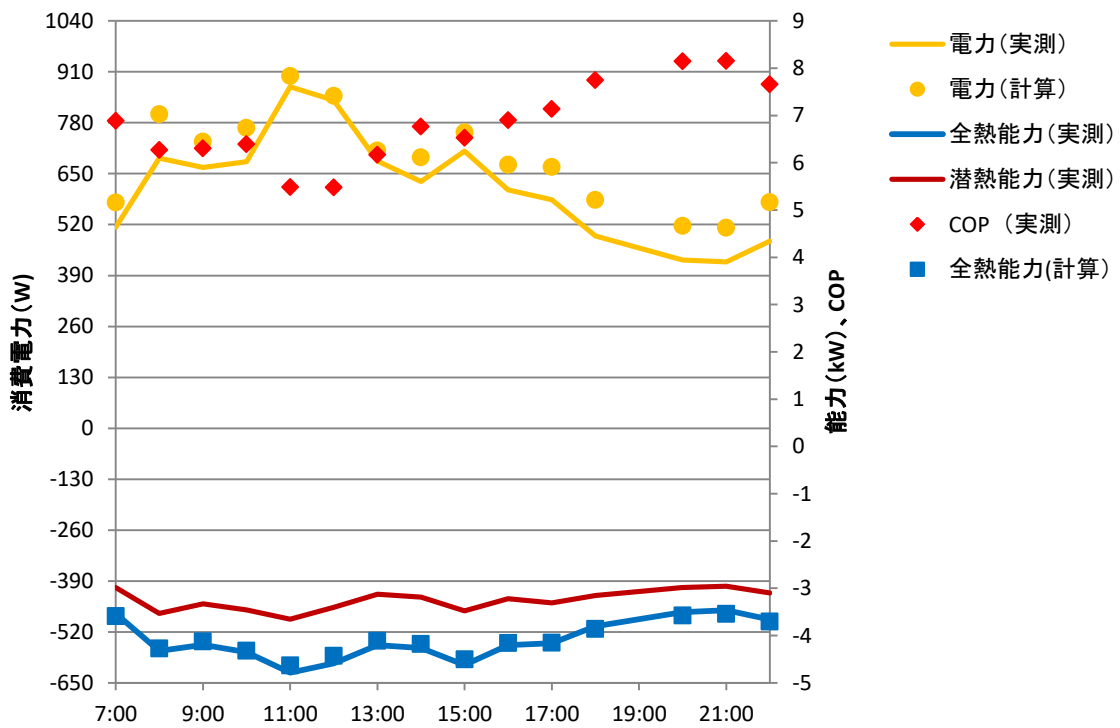


図 75 7/23 運転状態

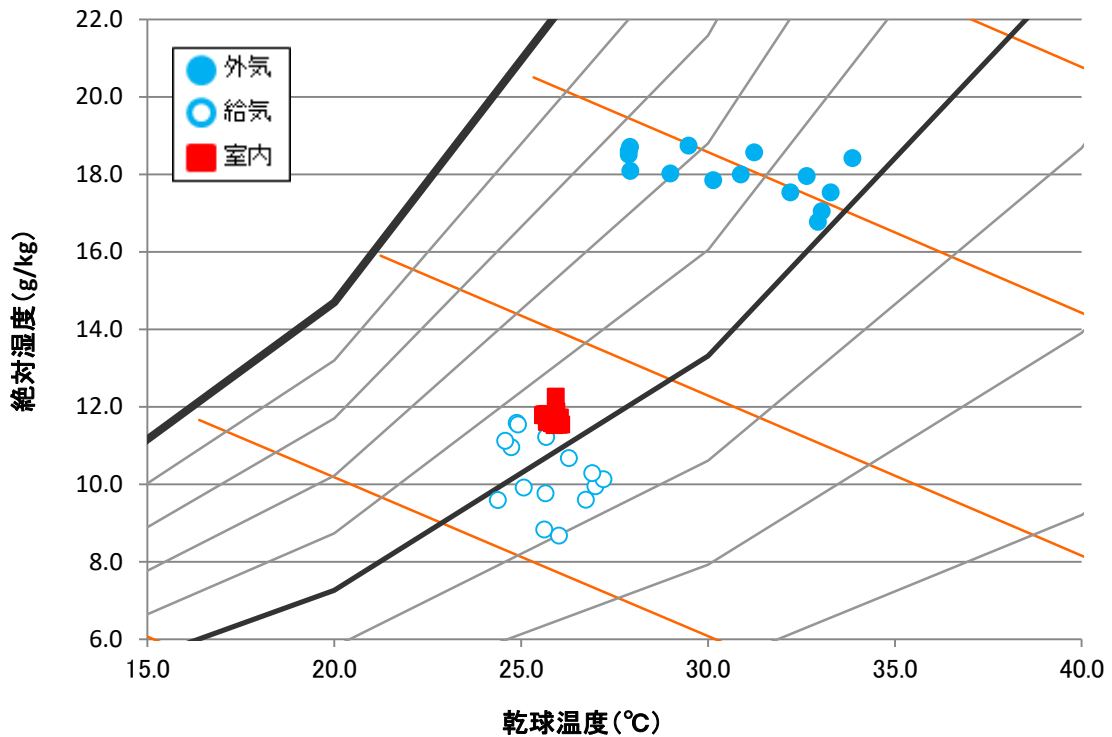


図 76 7/23 空気状態

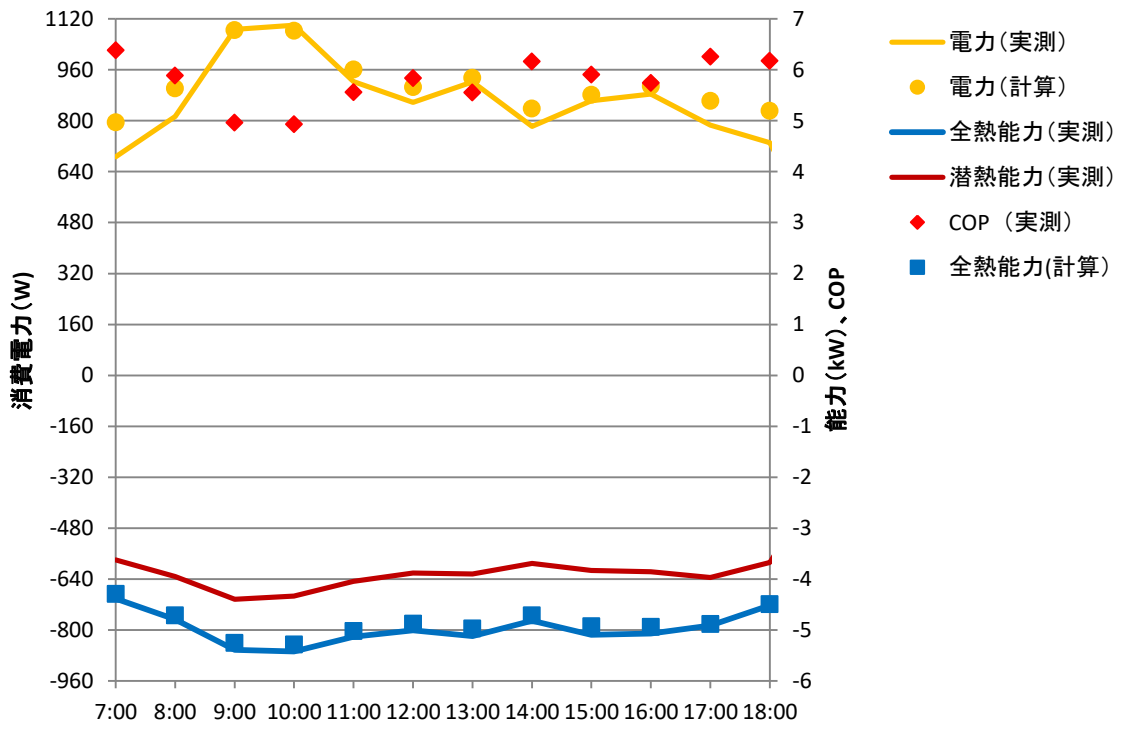


図 77 7/24 運転状態

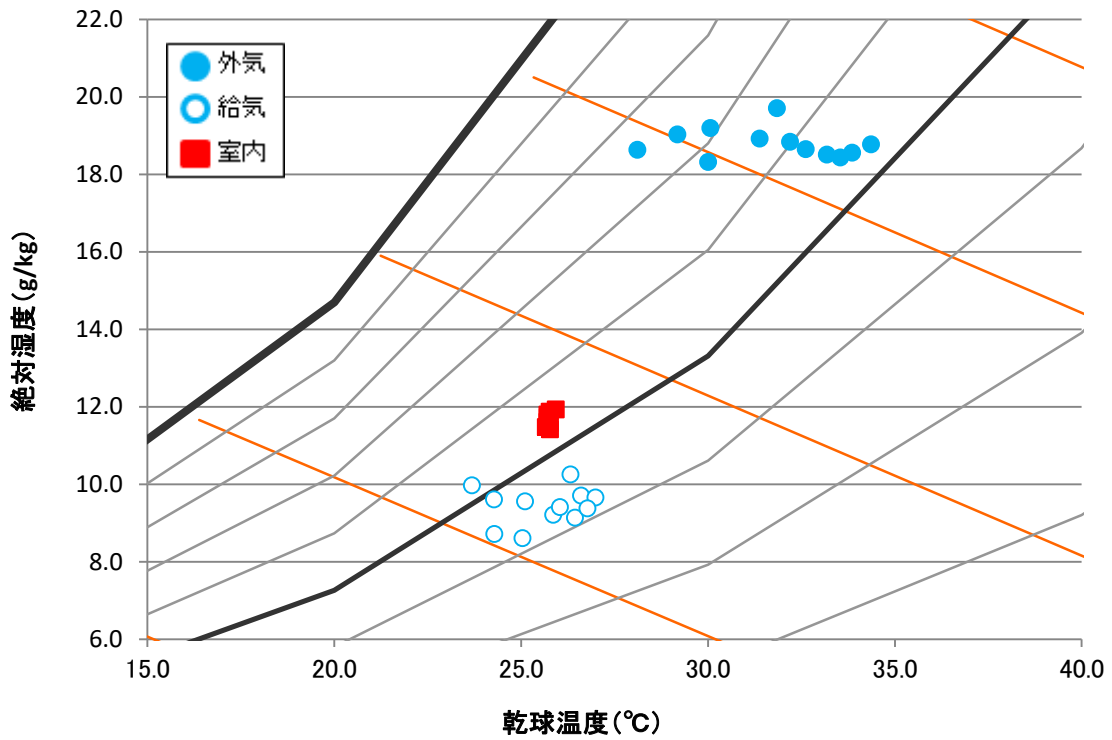


図 78 7/24 空気状態

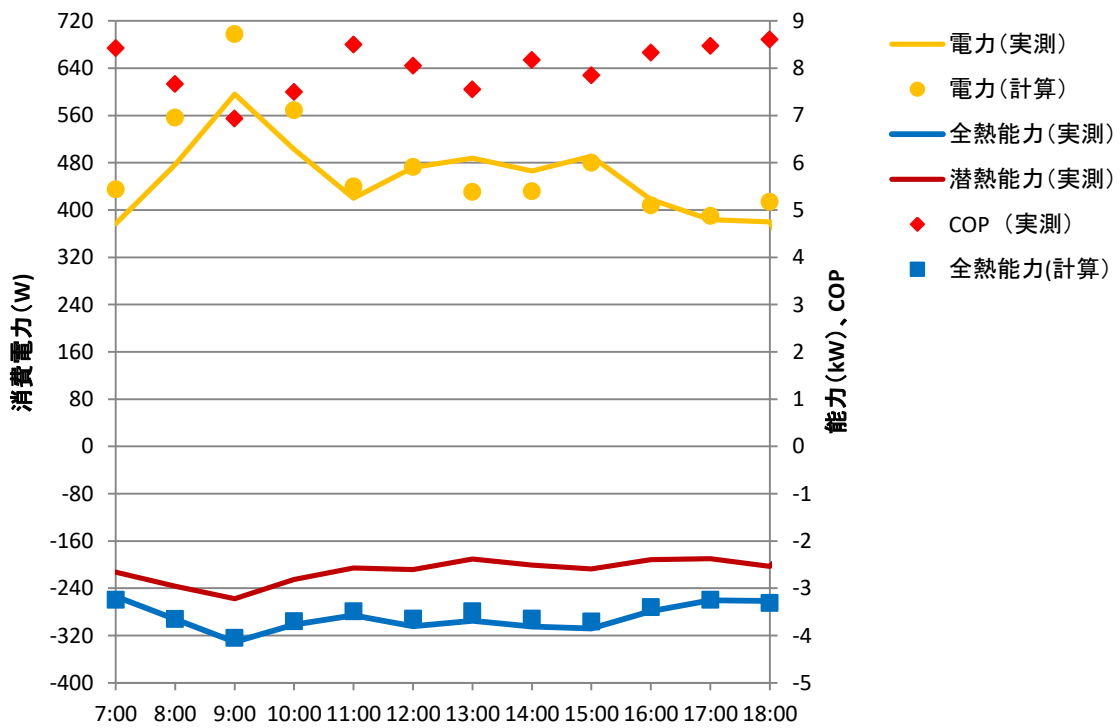


図 79 7/25 運転状態

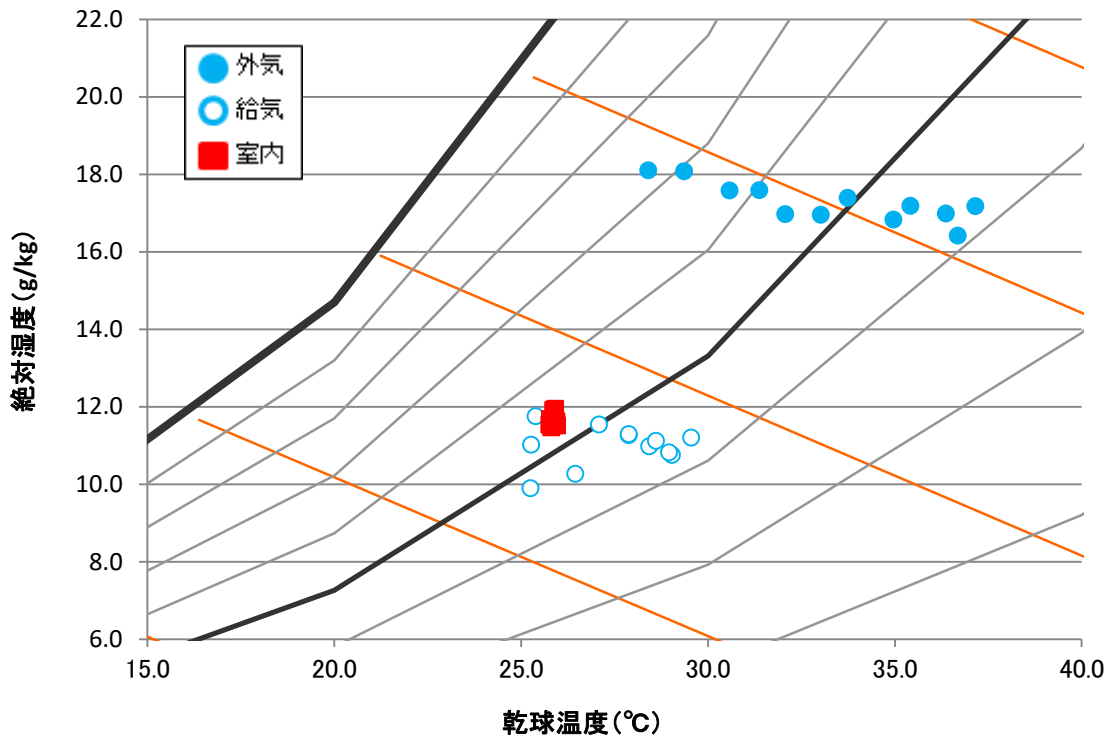


図 80 7/25 空気状態

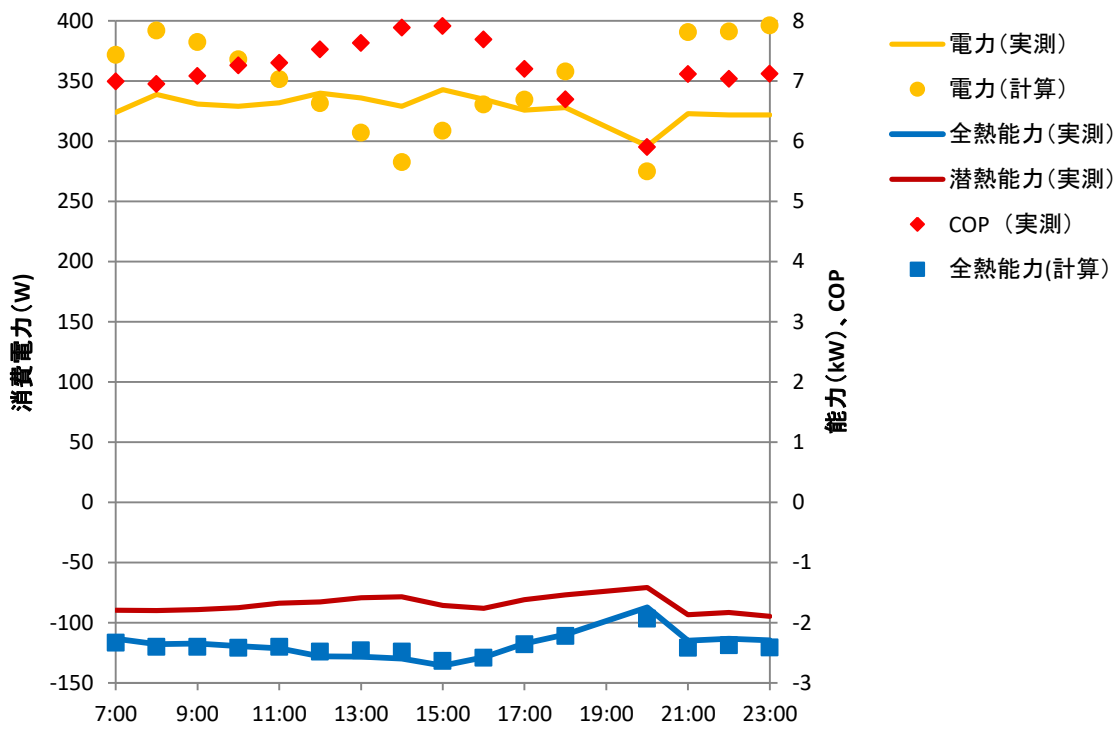


図 81 7/28 運転状態

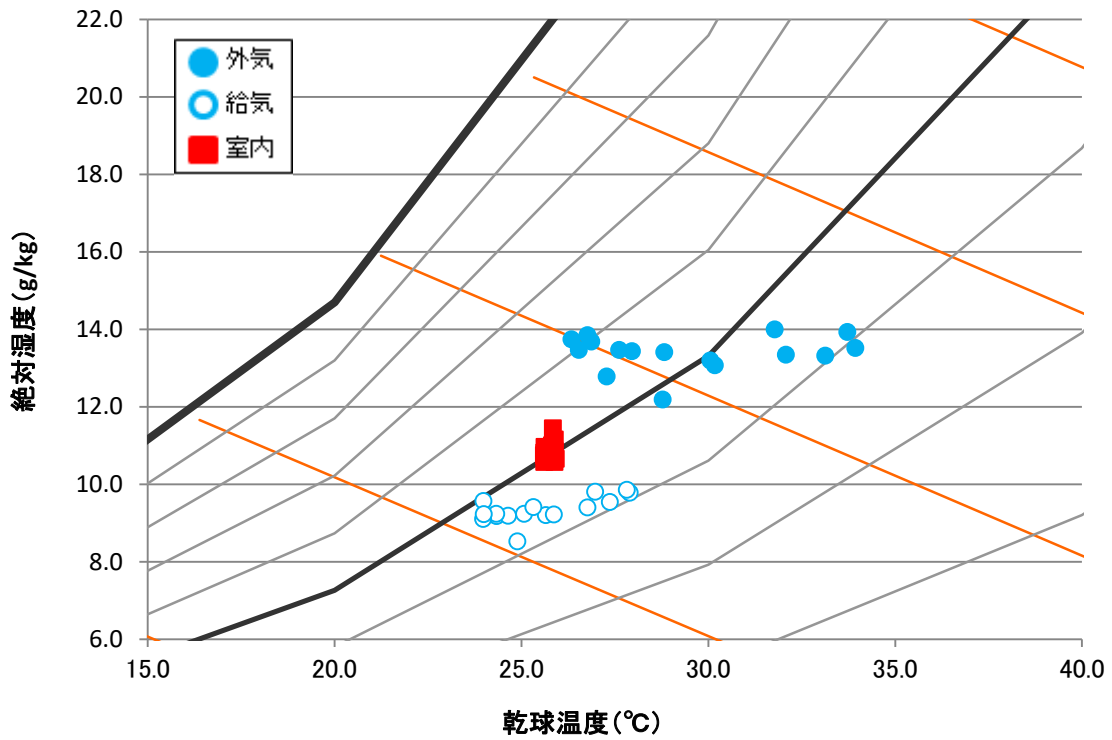


図 82 7/28 空気状態

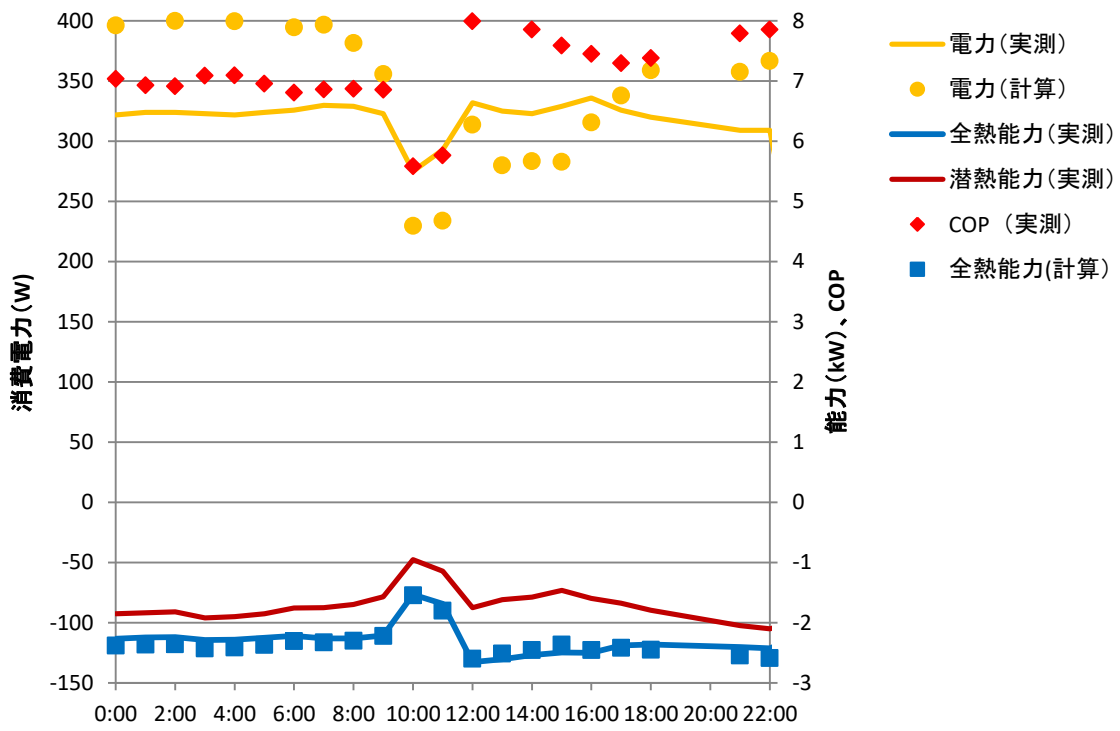


図 83 7/29 運転状態

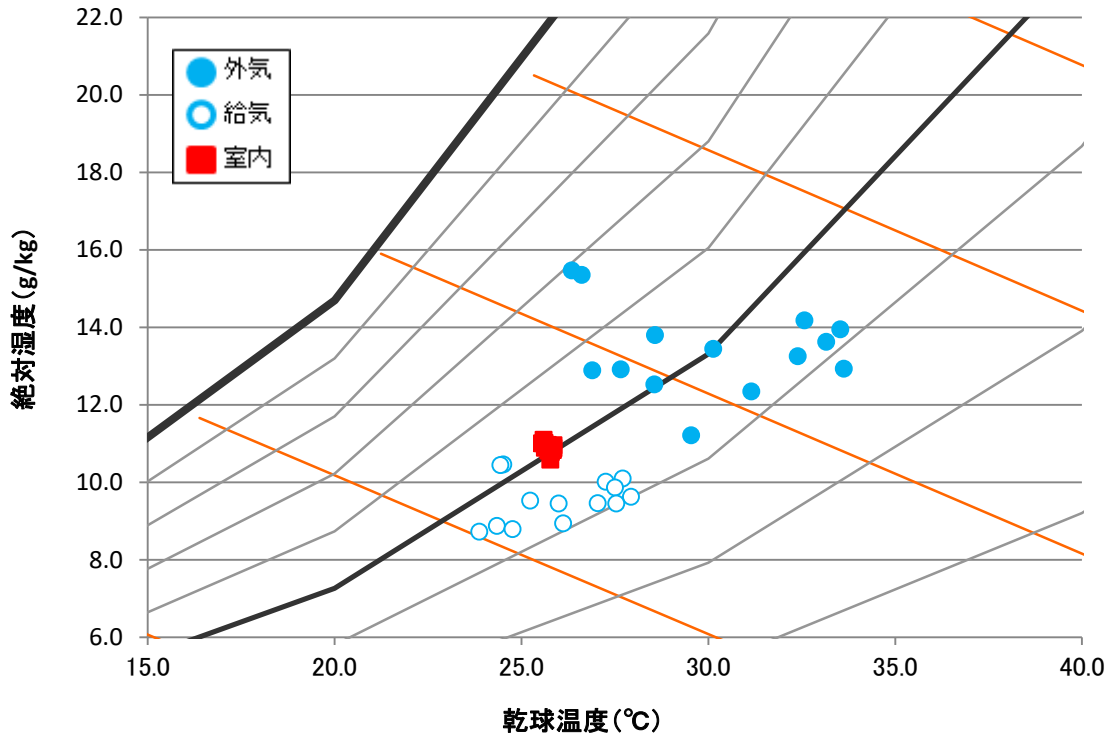


図 84 7/29 空気状態

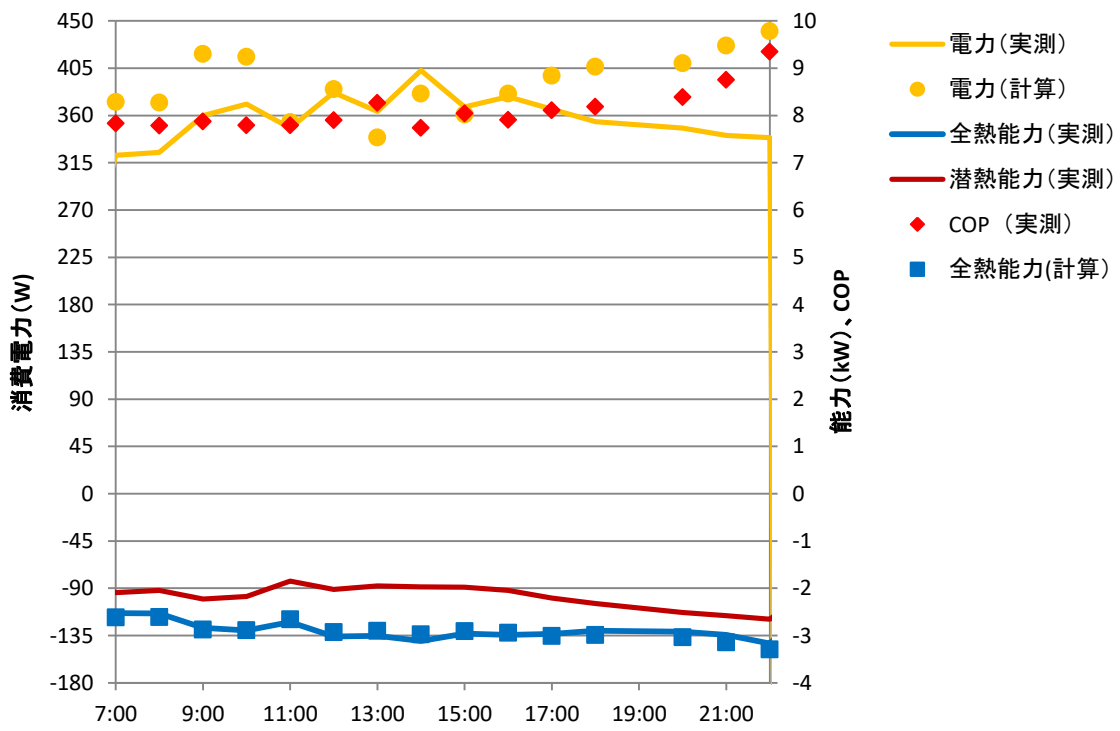


図 85 7/30 運転状態

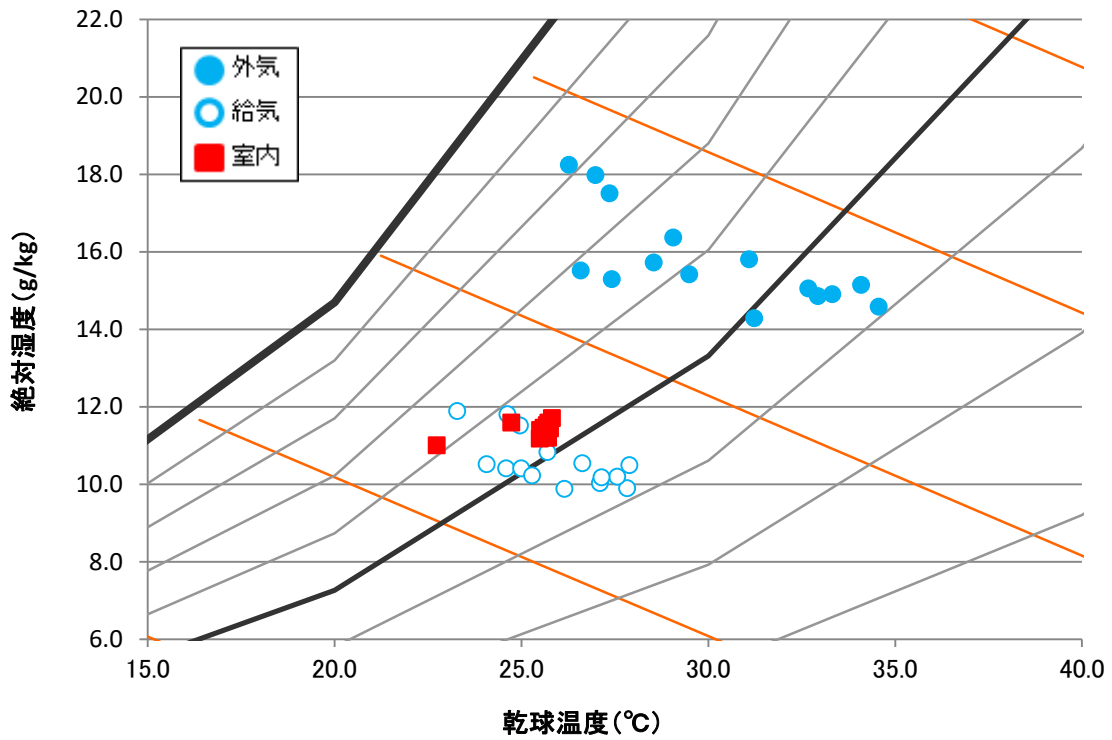


図 86 7/30 空気状態

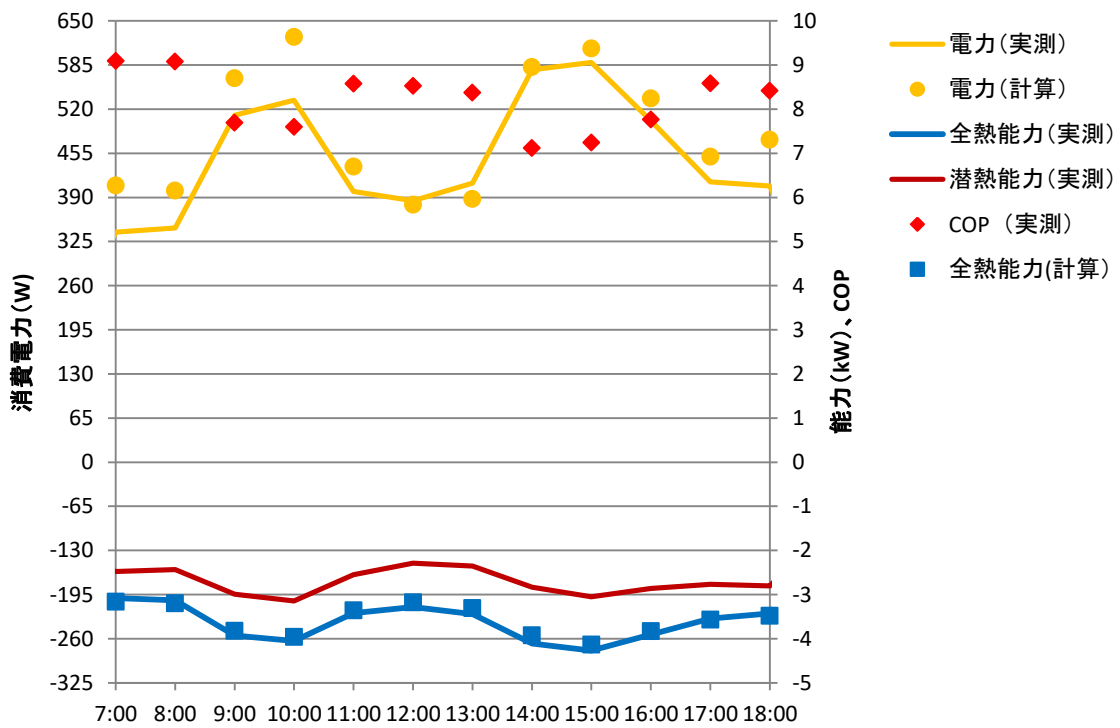


図 87 7/31 運転状態

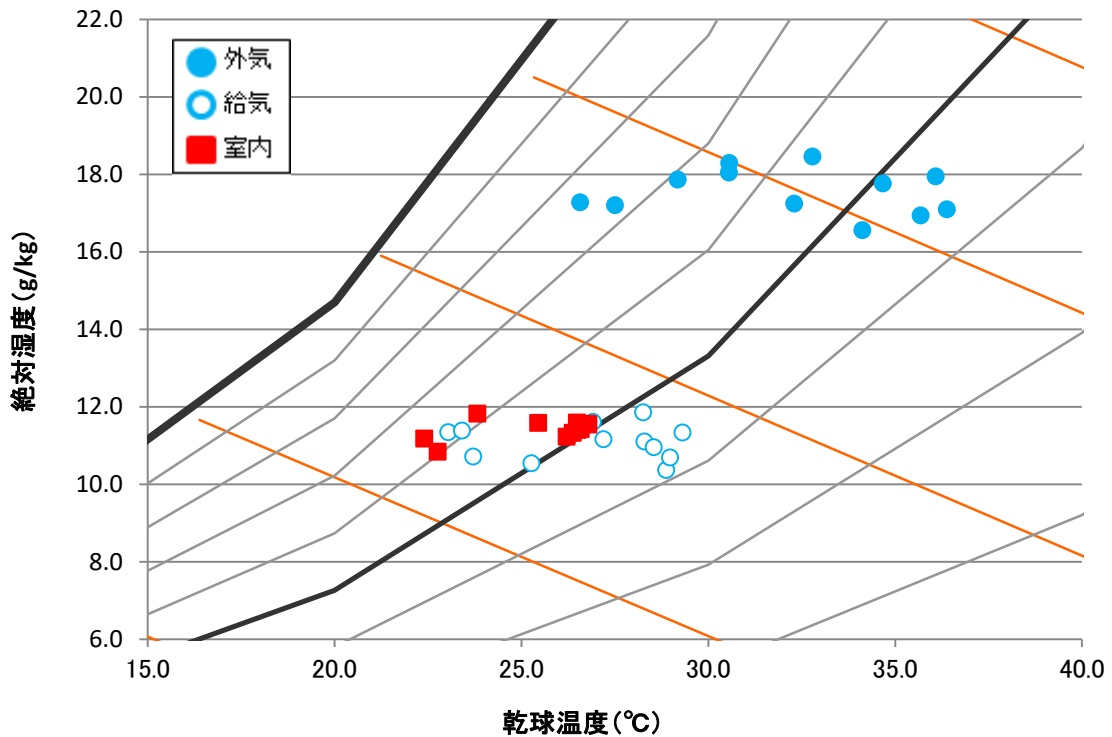


図 88 7/31 空気状態

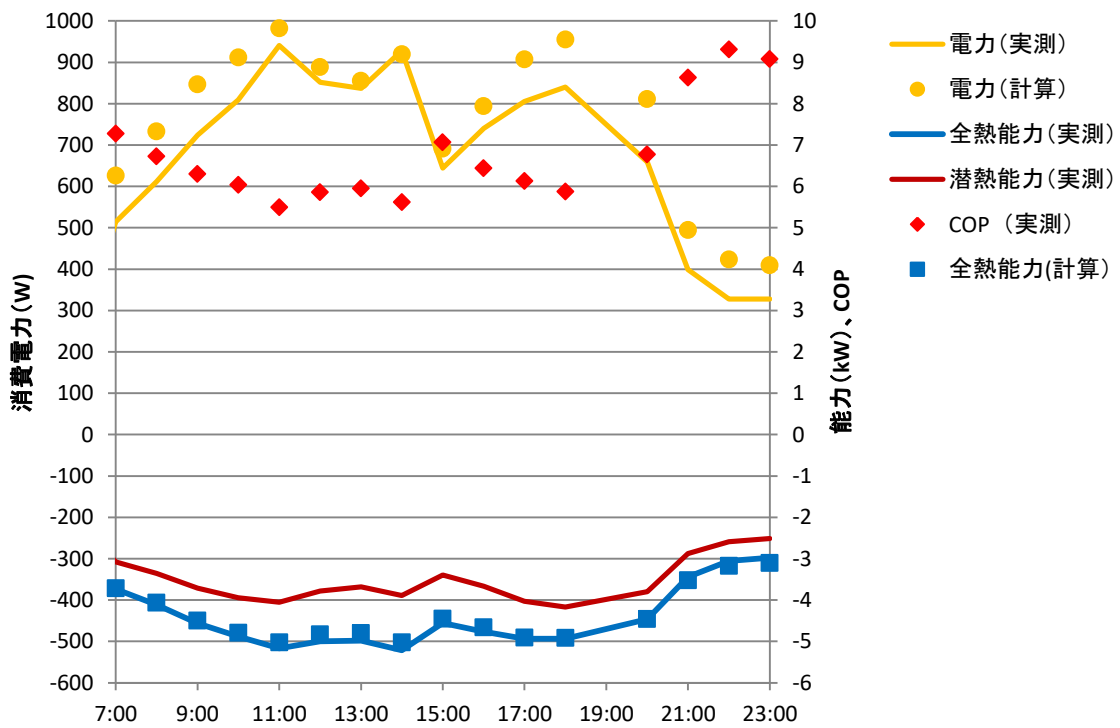


図 89 8/1 運転状態

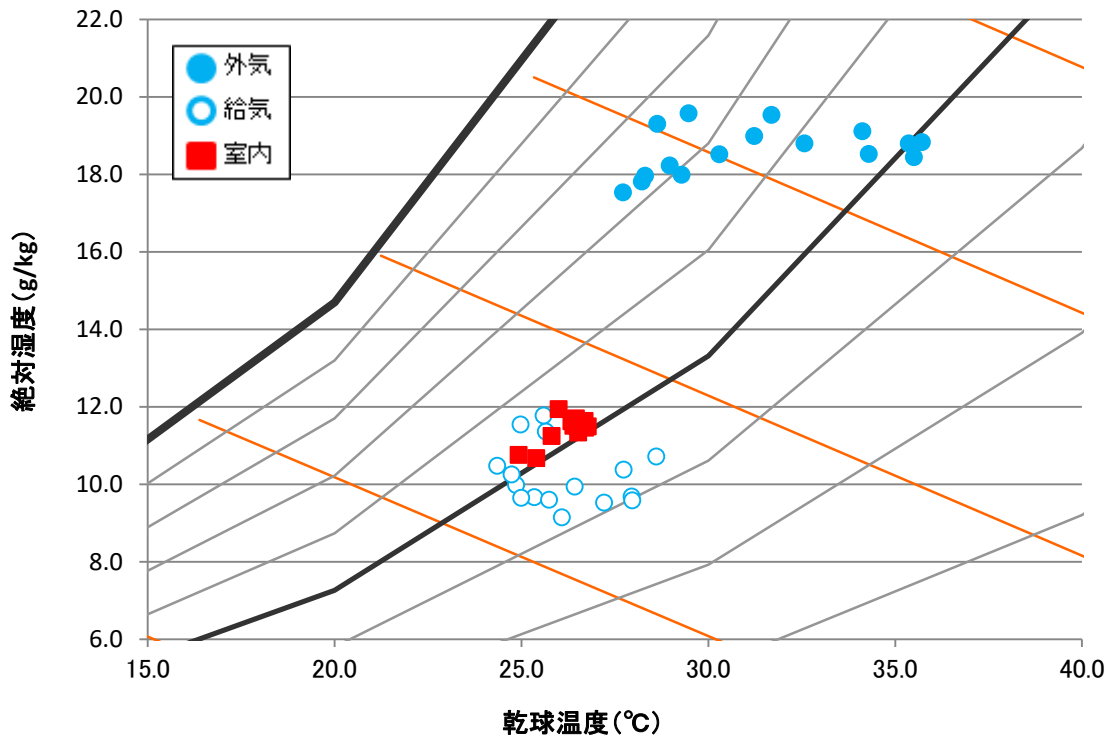


図 90 8/1 空気状態

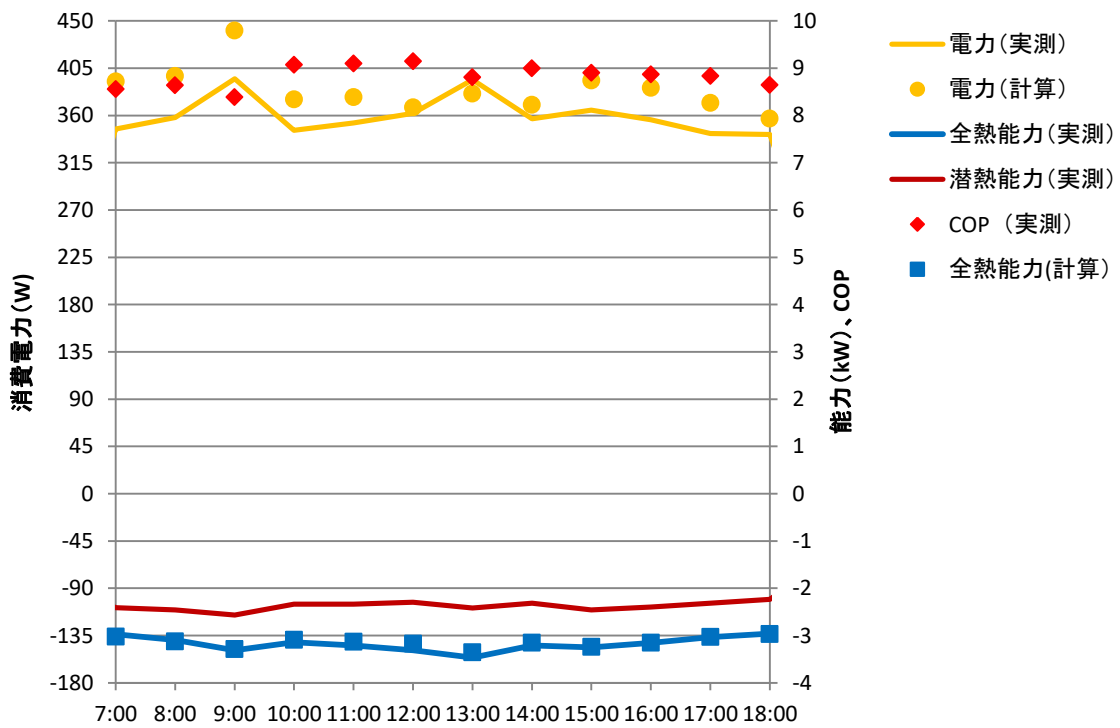


図 91 8/4 運転状態

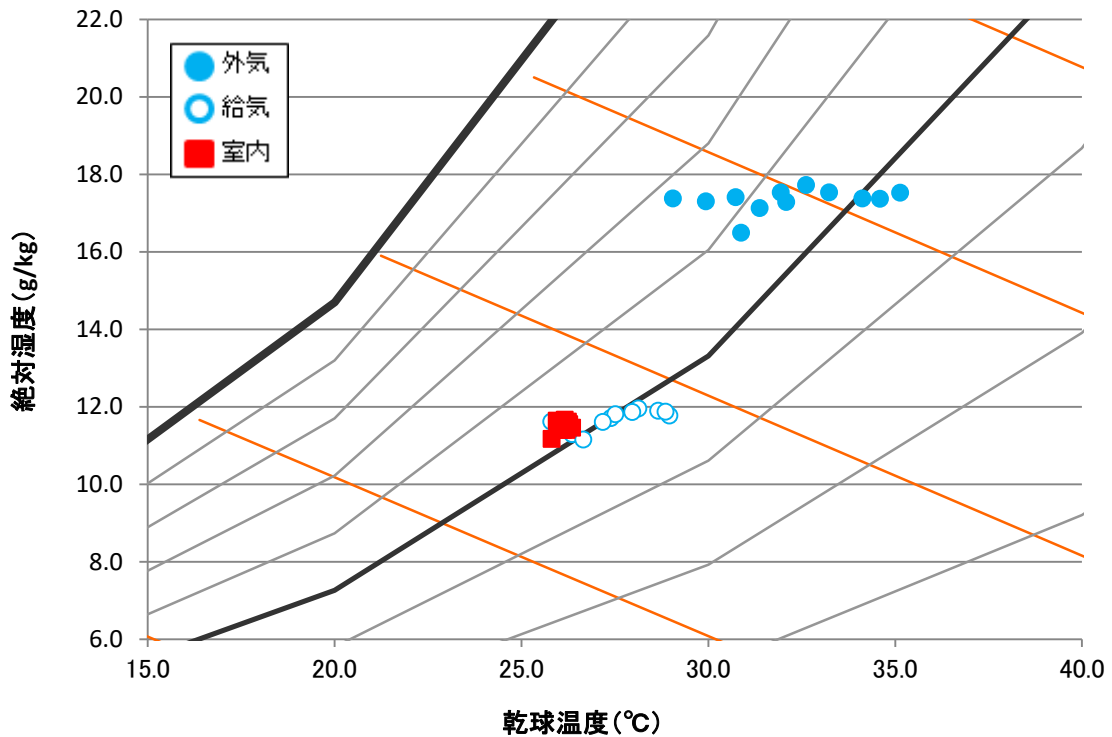


図 92 8/4 空気状態

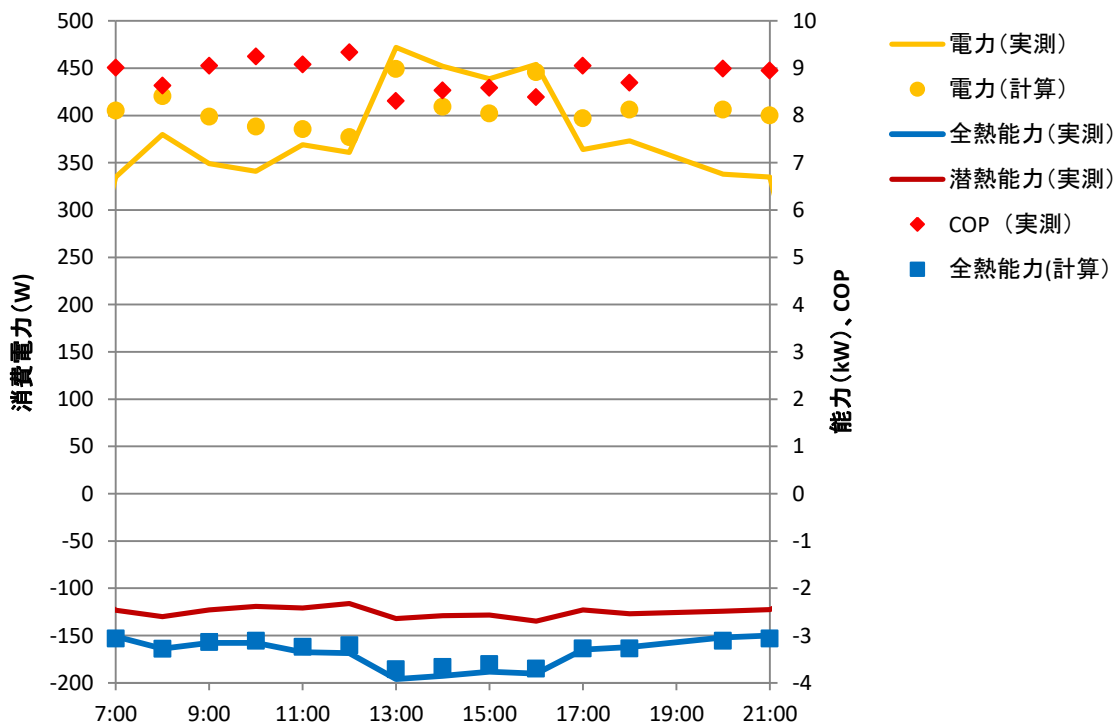


図 93 8/5 運転状態

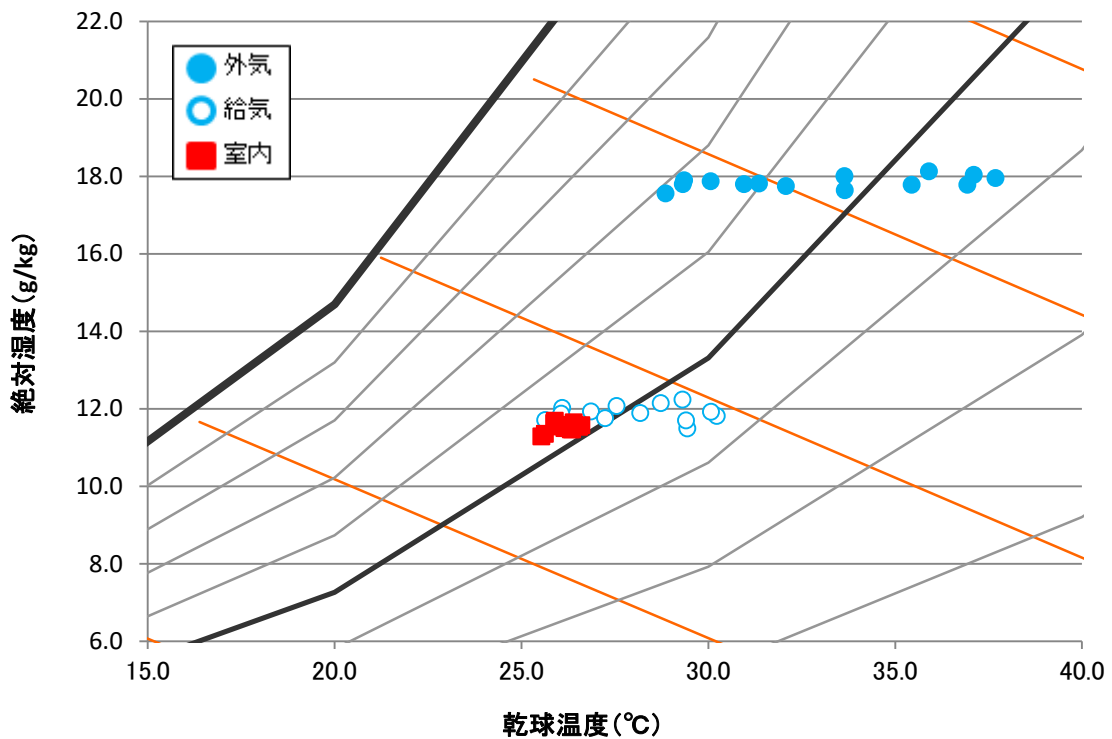


図 94 8/5 空気状態

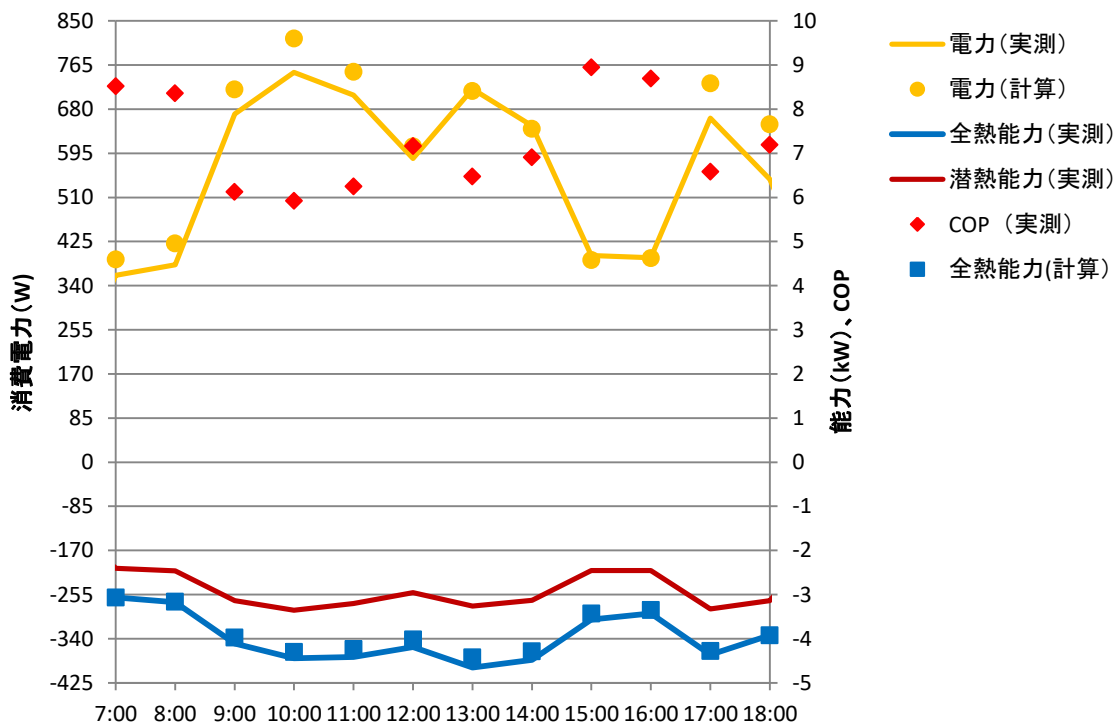


図 95 8/6 運転状態

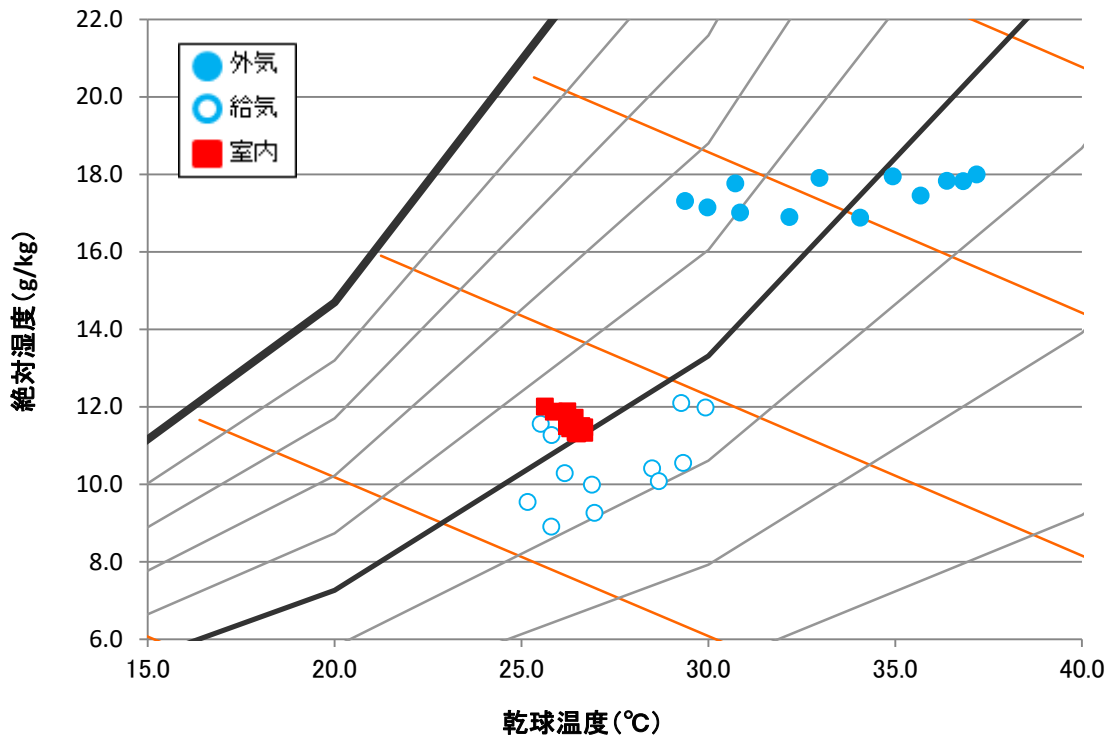


図 96 8/6 空気状態

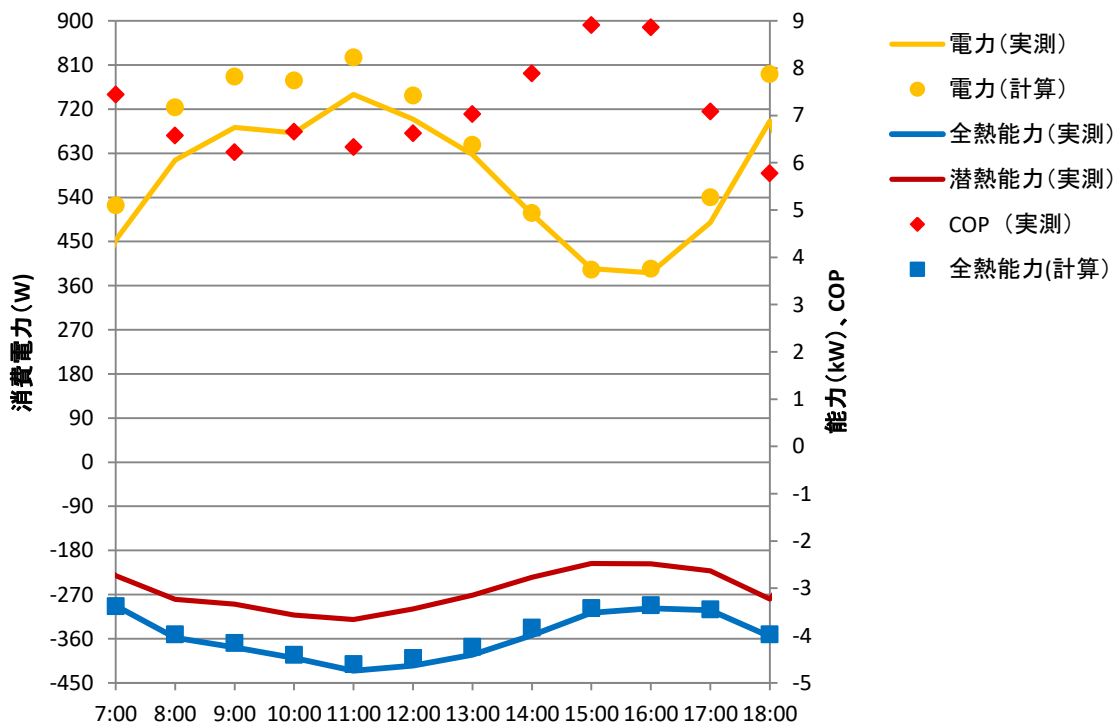


図 97 8/7 運転状態

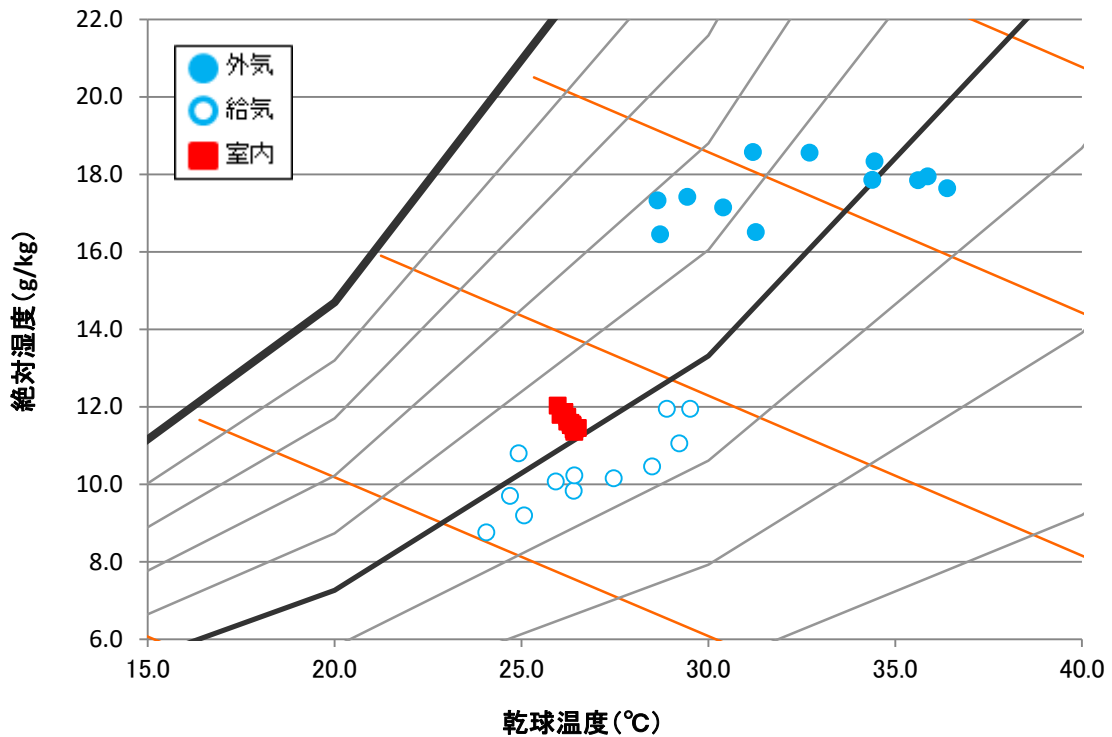


図 98 8/7 空気状態

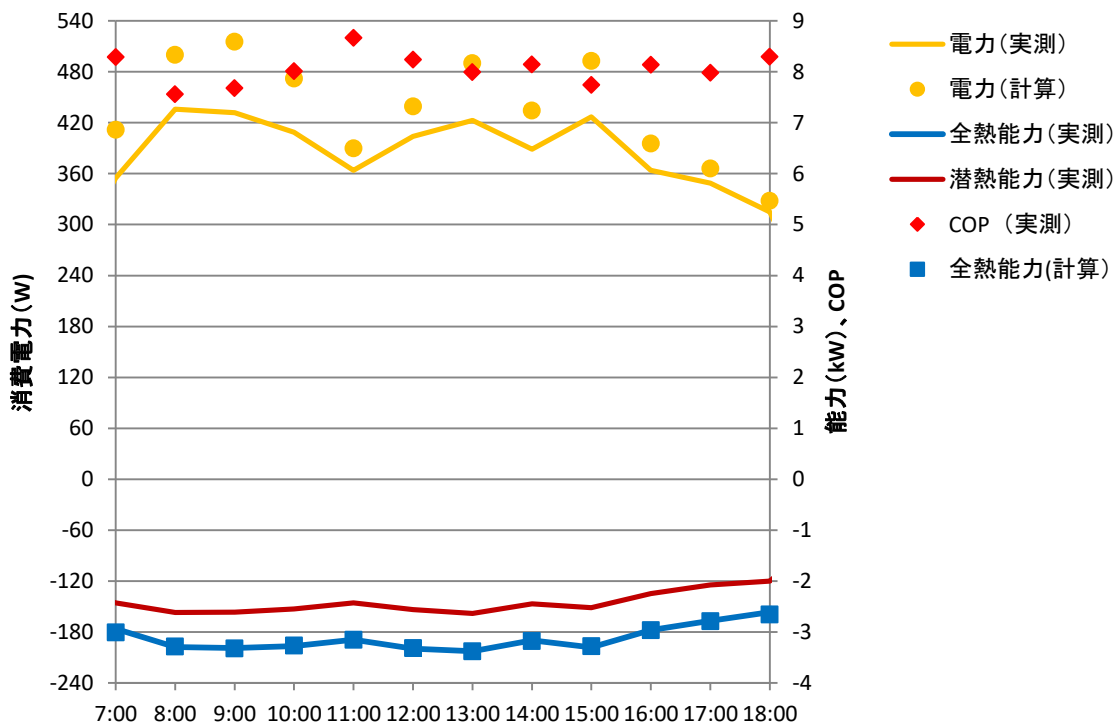


図 99 8/8 運転状態

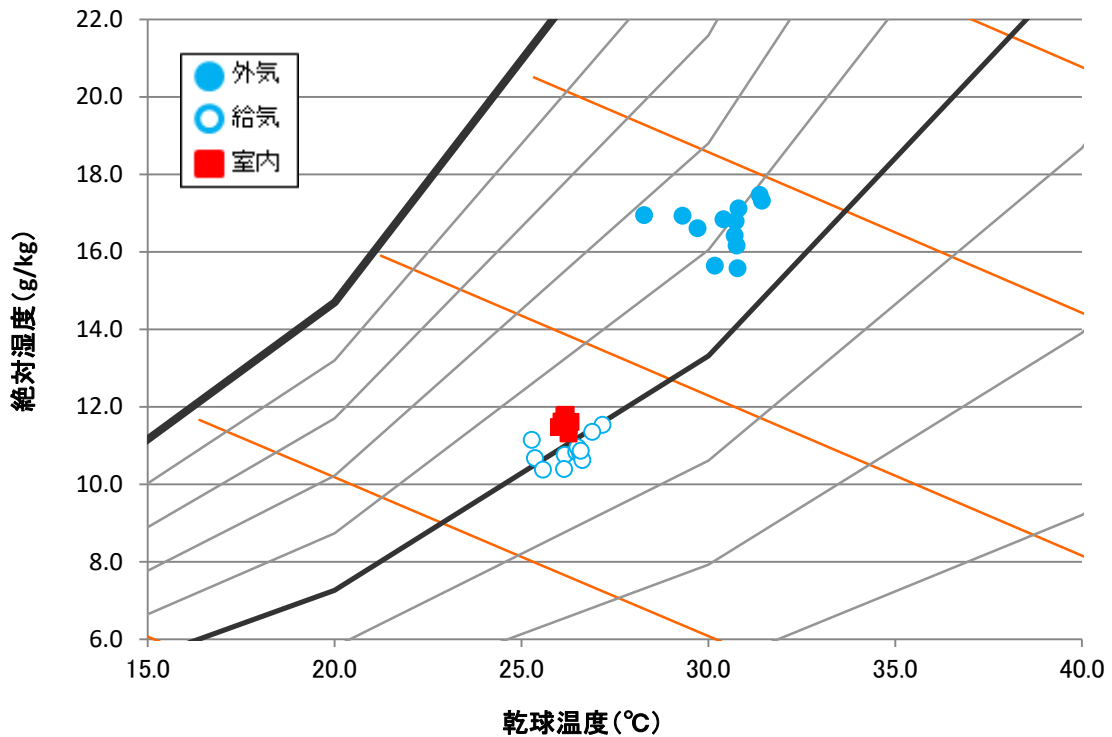


図 100 8/8 空気状態

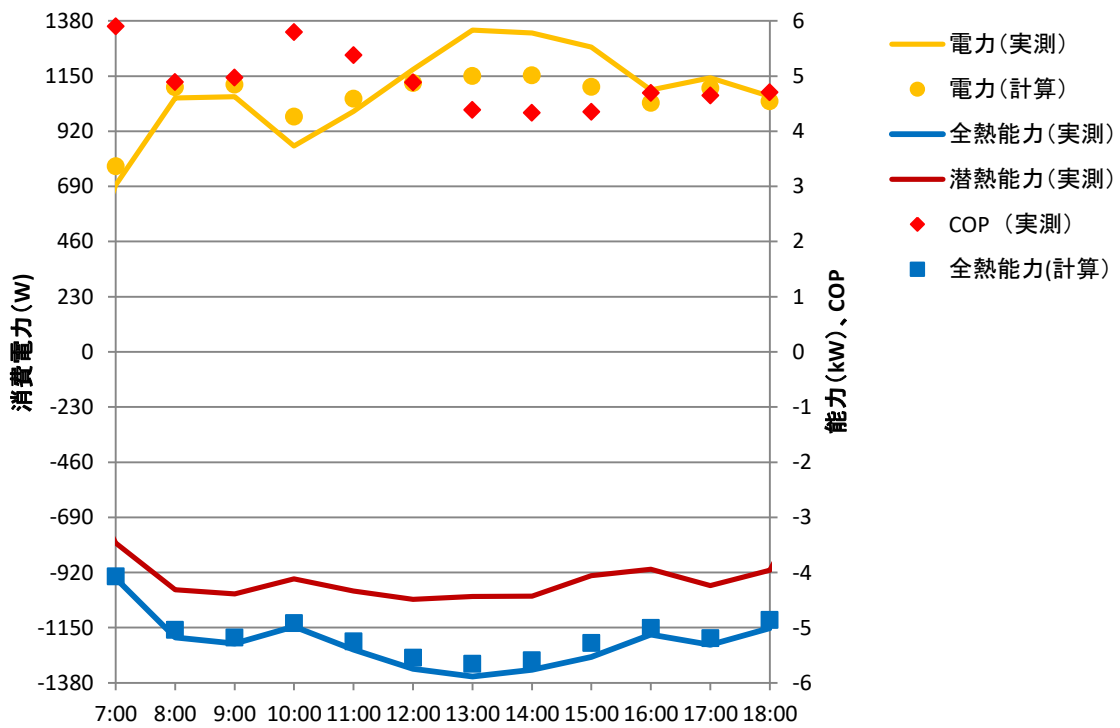


図 101 8/18 運転状態

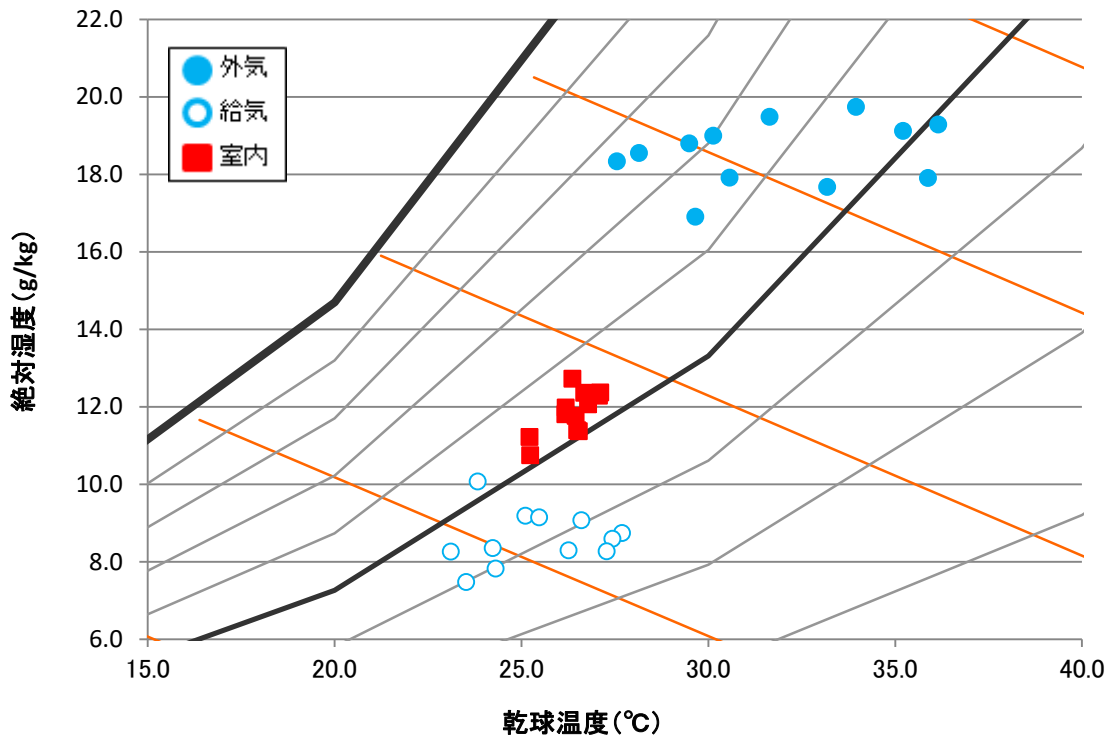


図 102 8/18 空気状態

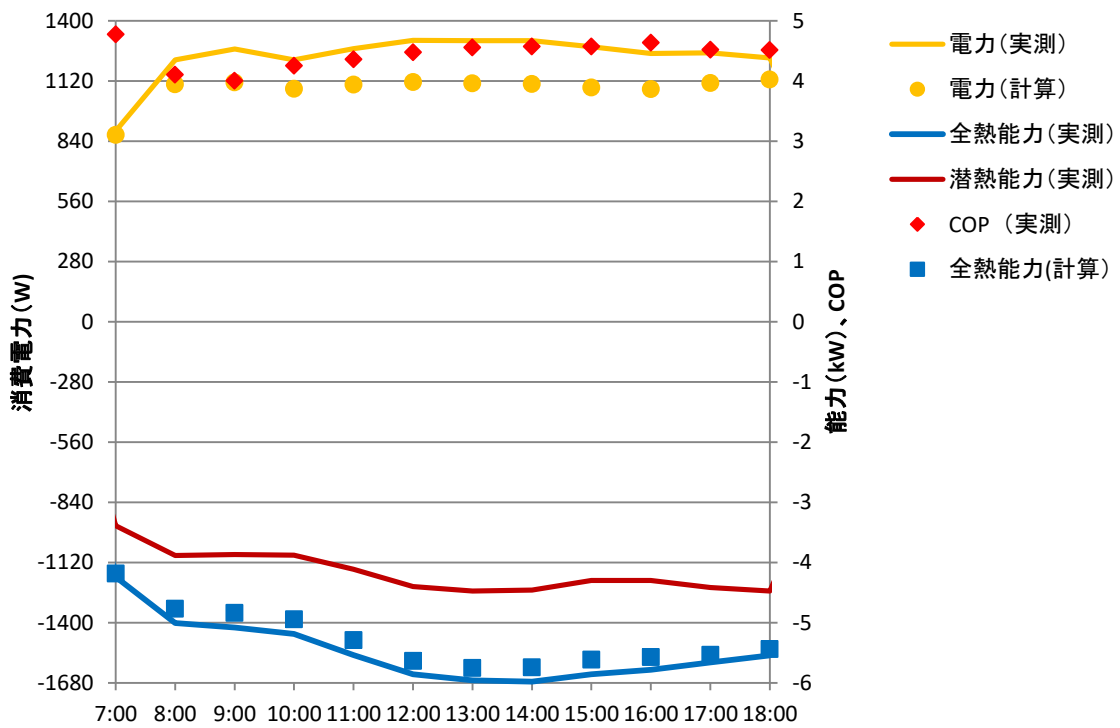


図 103 8/19 運転状態

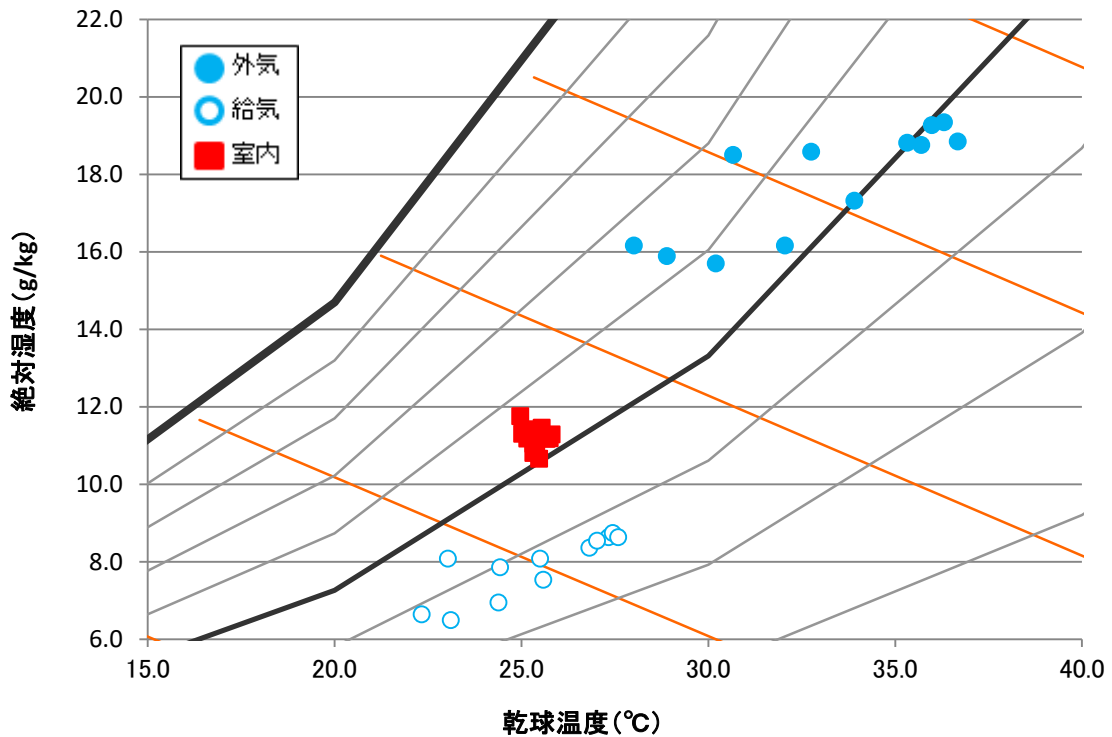


図 104 8/19 空気状態

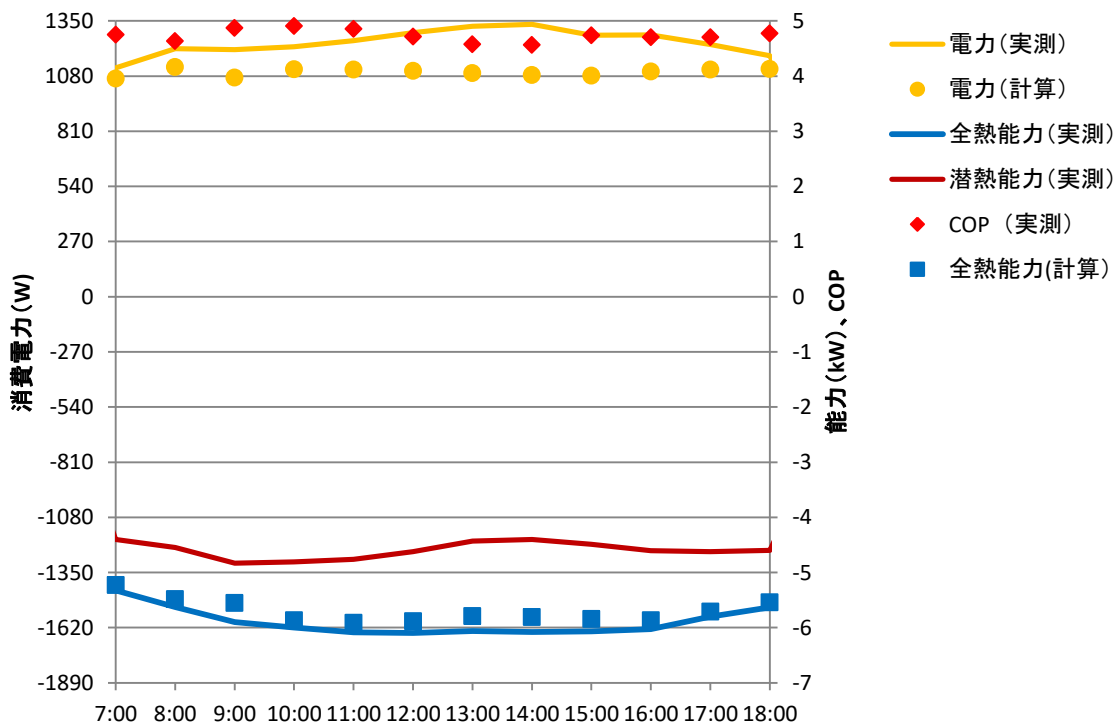


図 105 8/20 運転状態

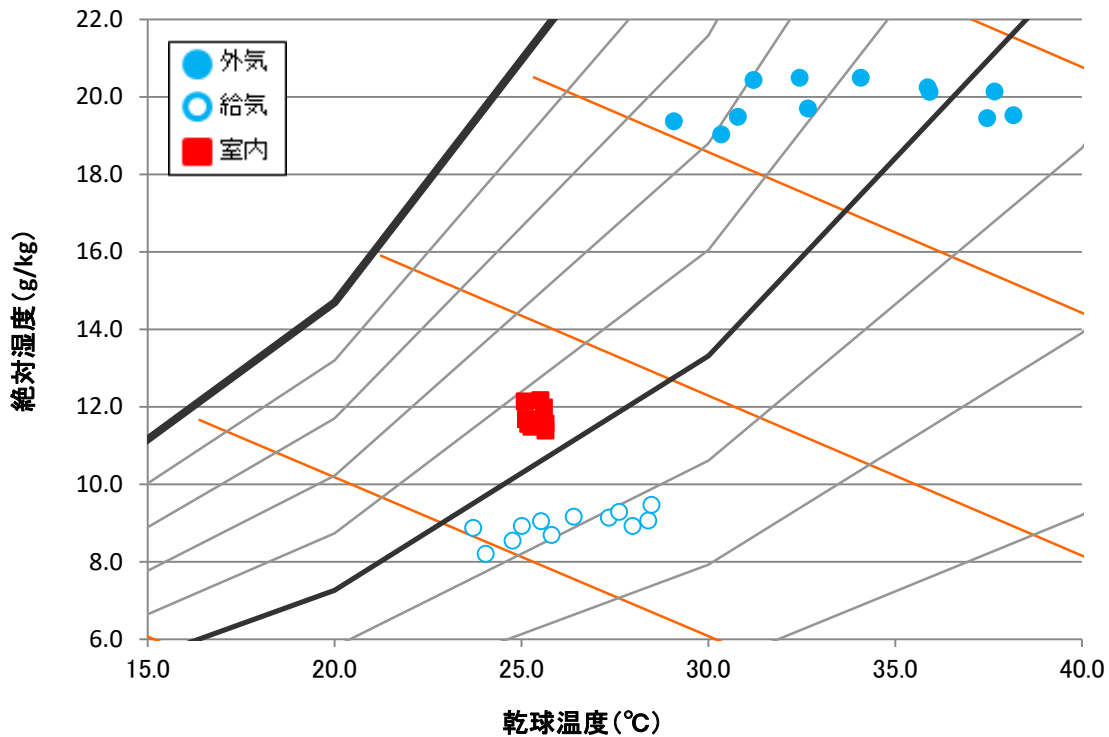


図 106 8/20 空気状態

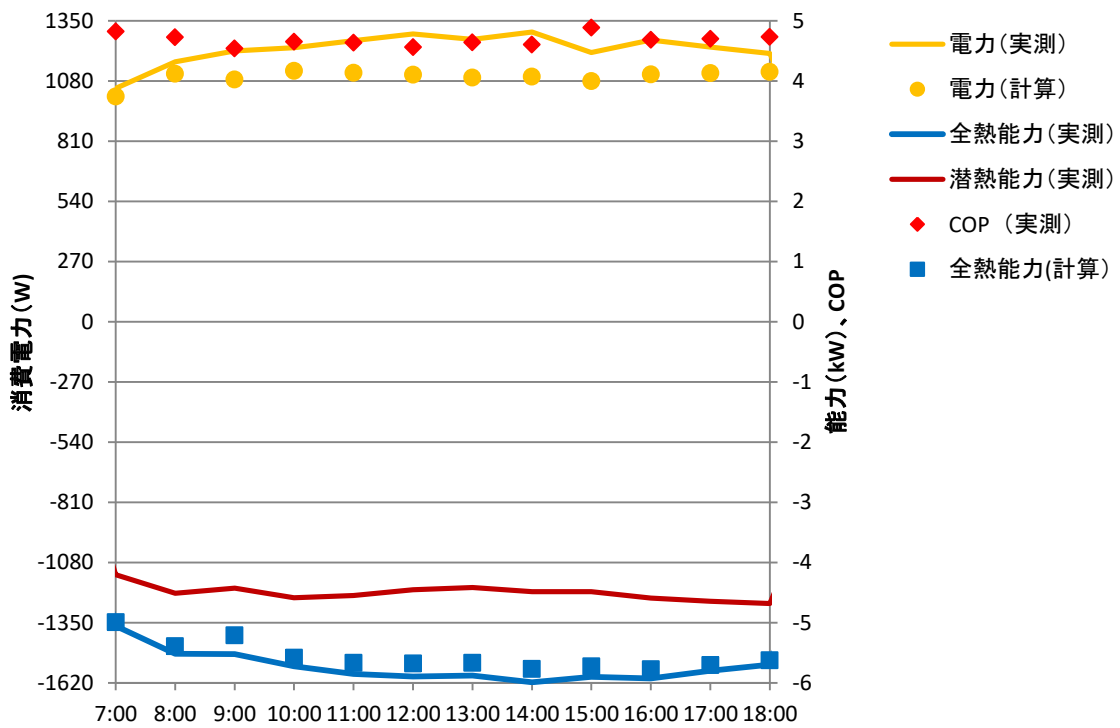


図 107 8/21 運転状態

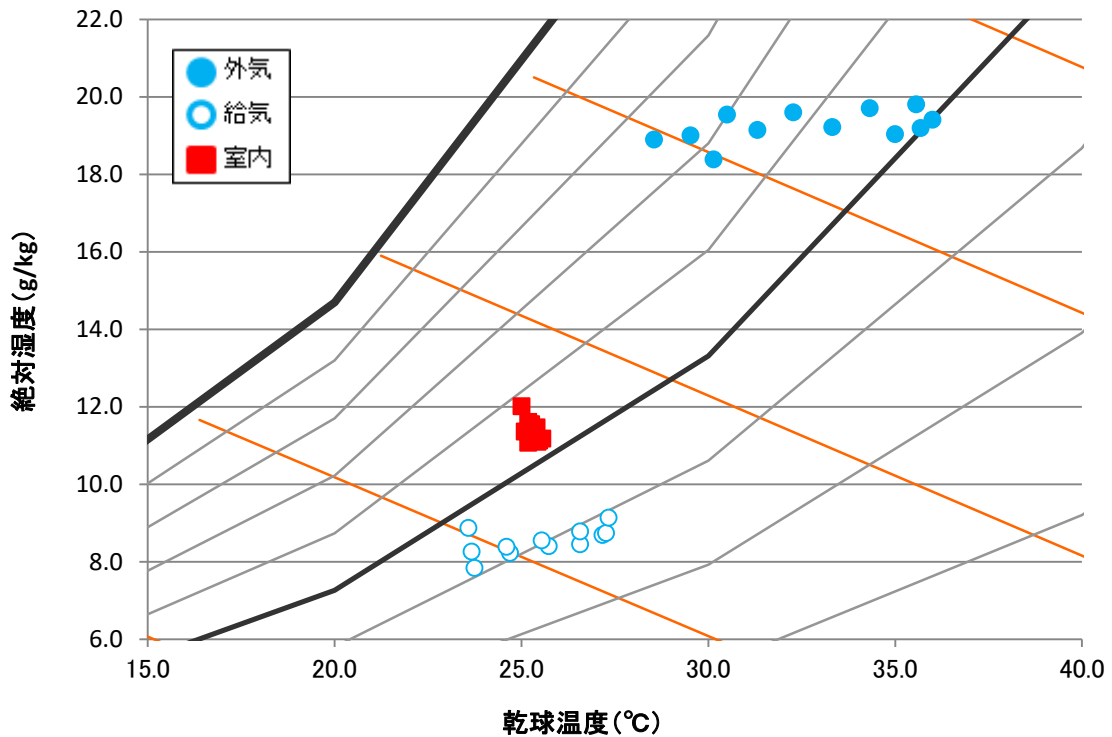


図 108 8/21 空気状態

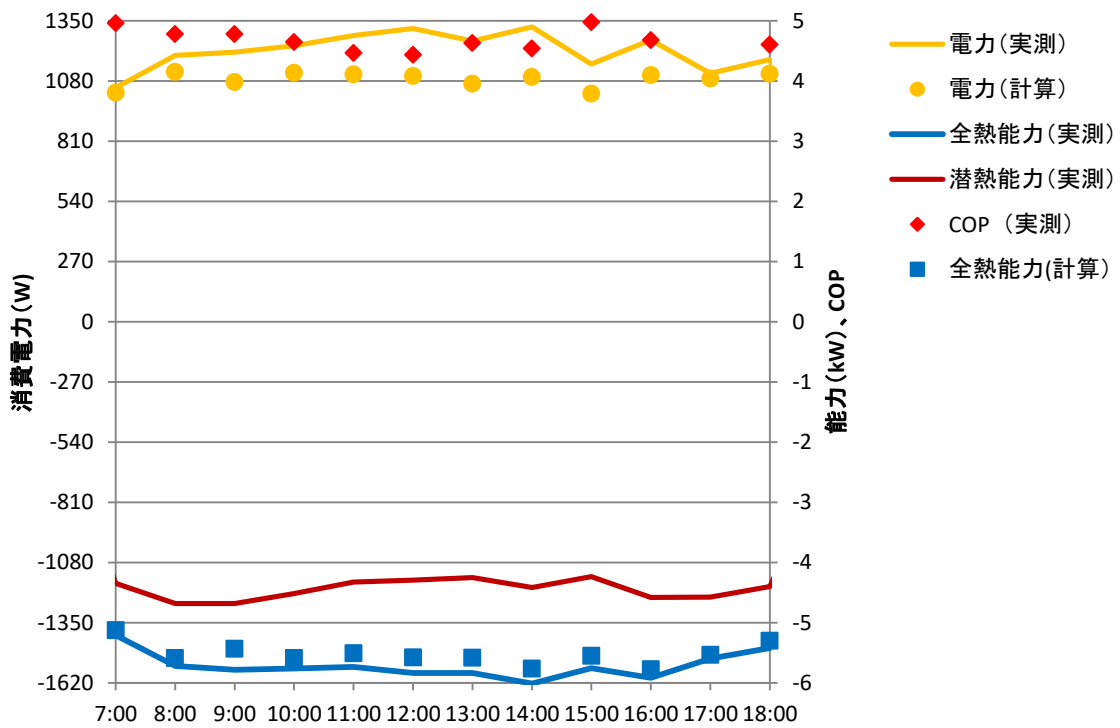


図 109 8/22 運転状態

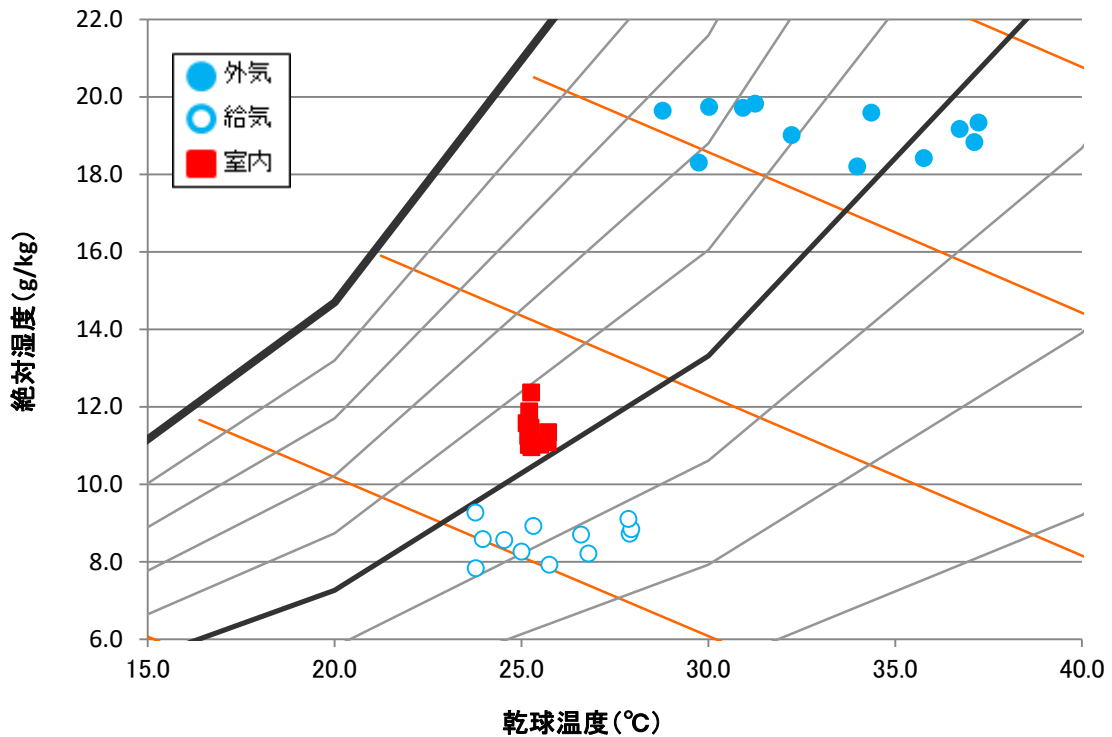


図 110 8/22 空気状態

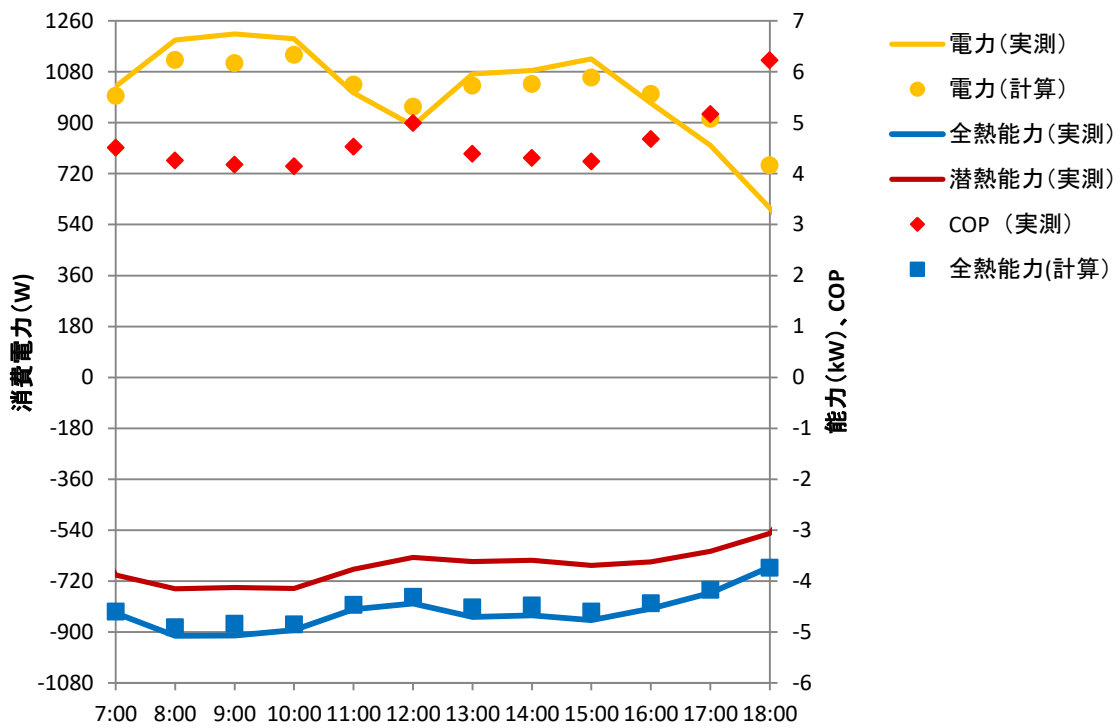


图 111 8/25 运转状态

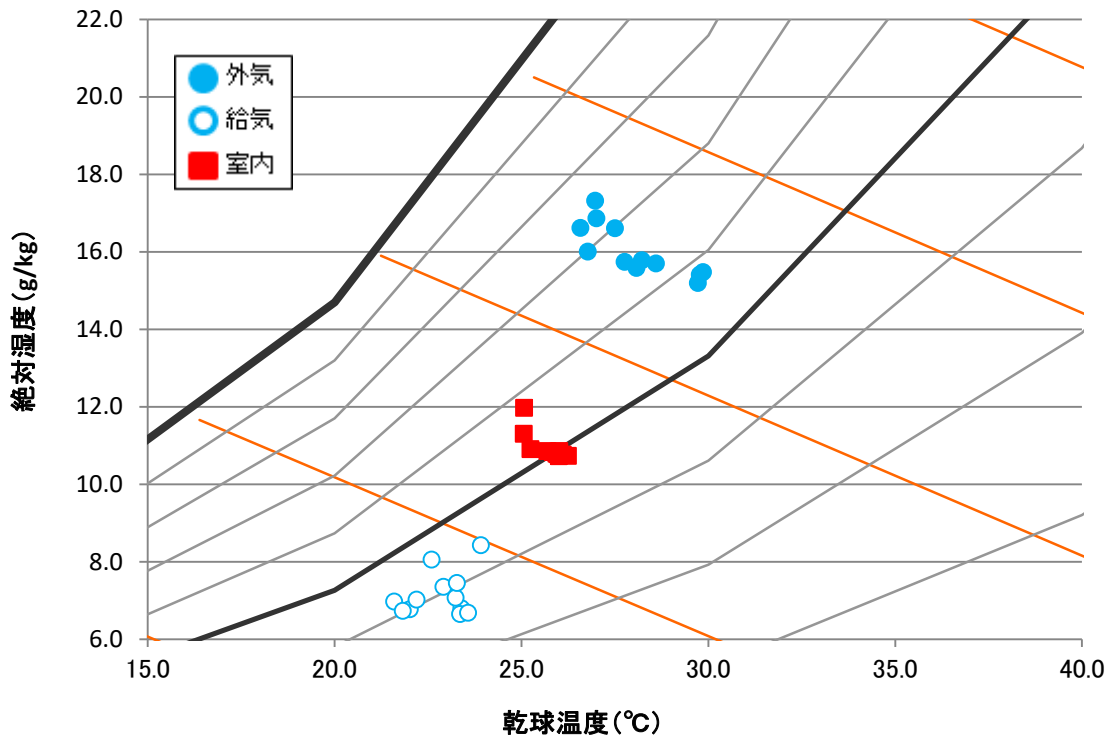


图 112 8/25 空气状态

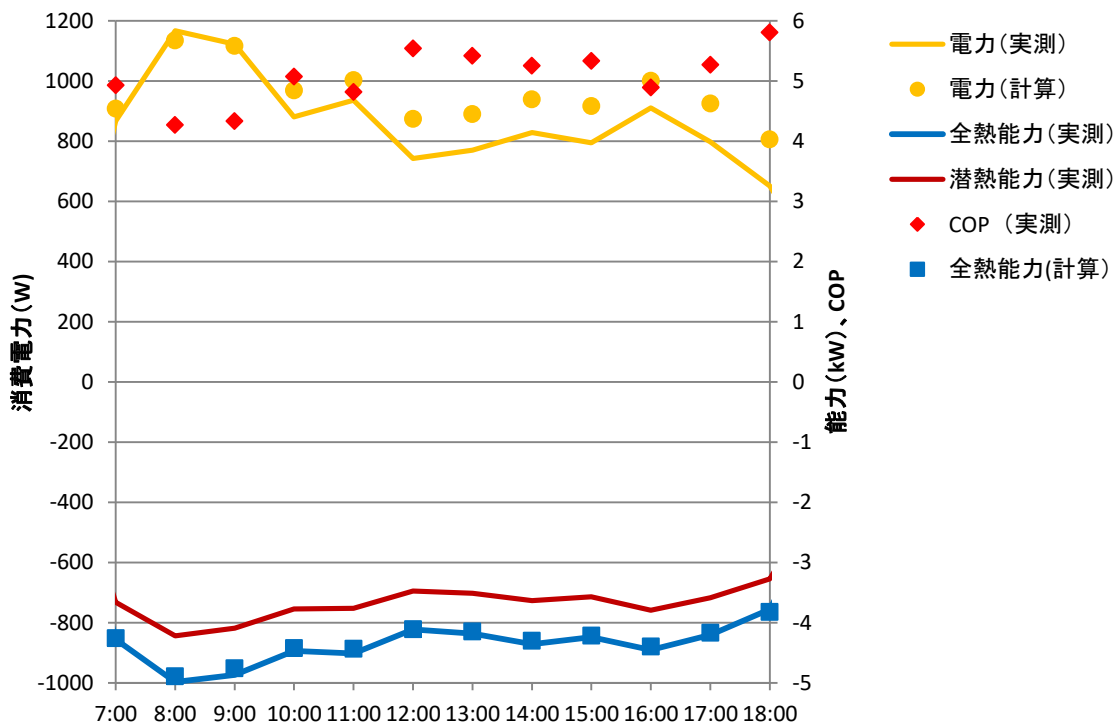


図 113 8/26 運転状態

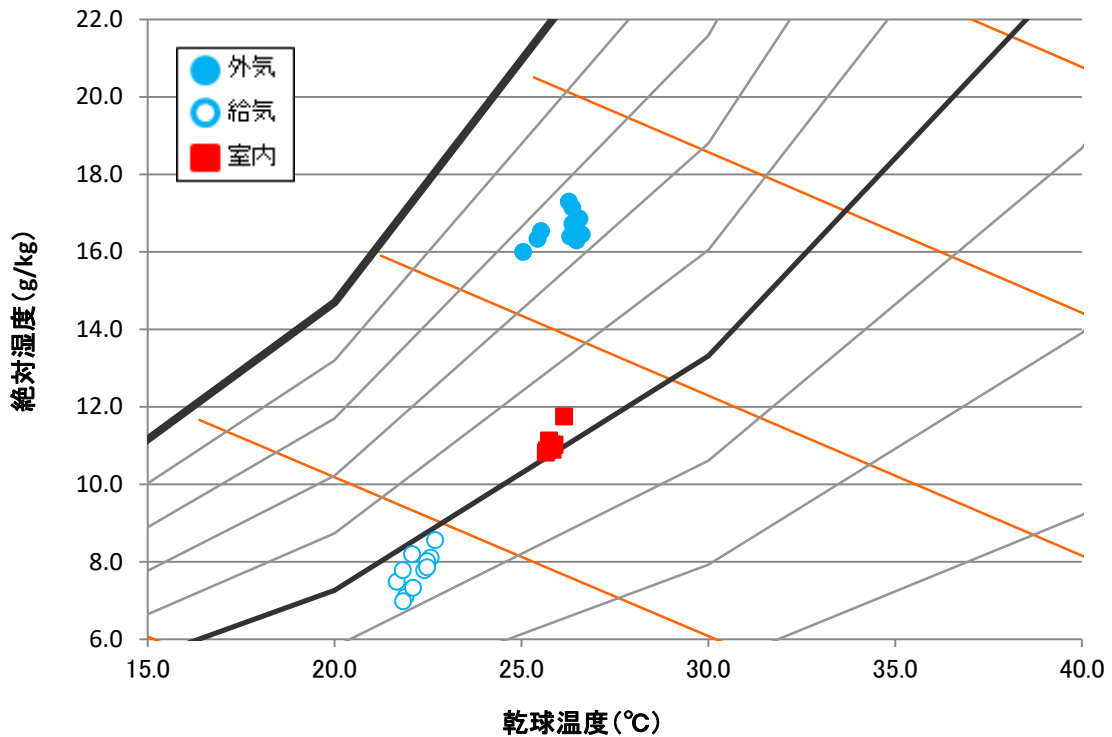


図 114 8/26 空気状態

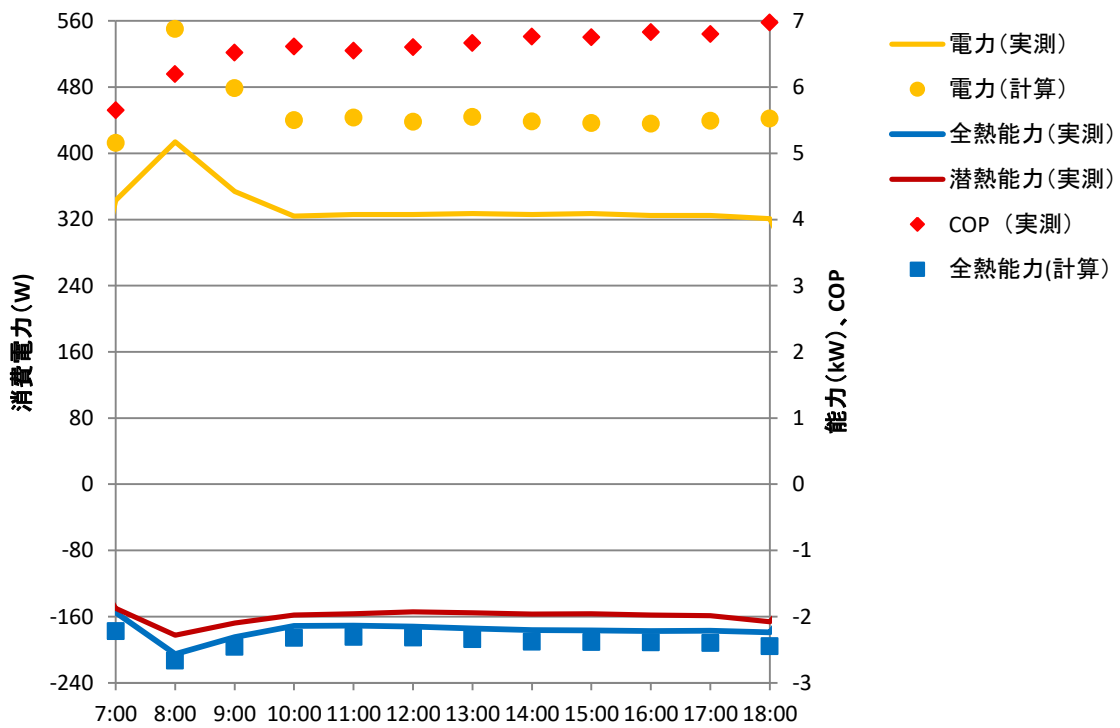


図 115 8/27 運転状態

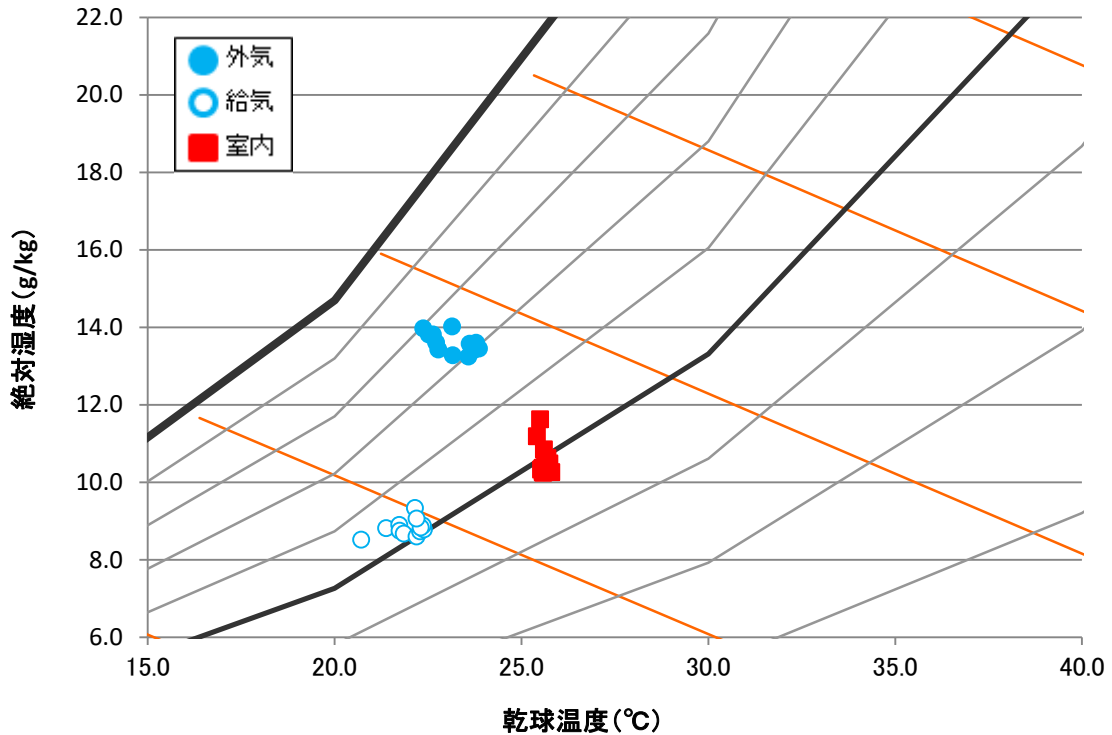


図 116 8/27 空気状態

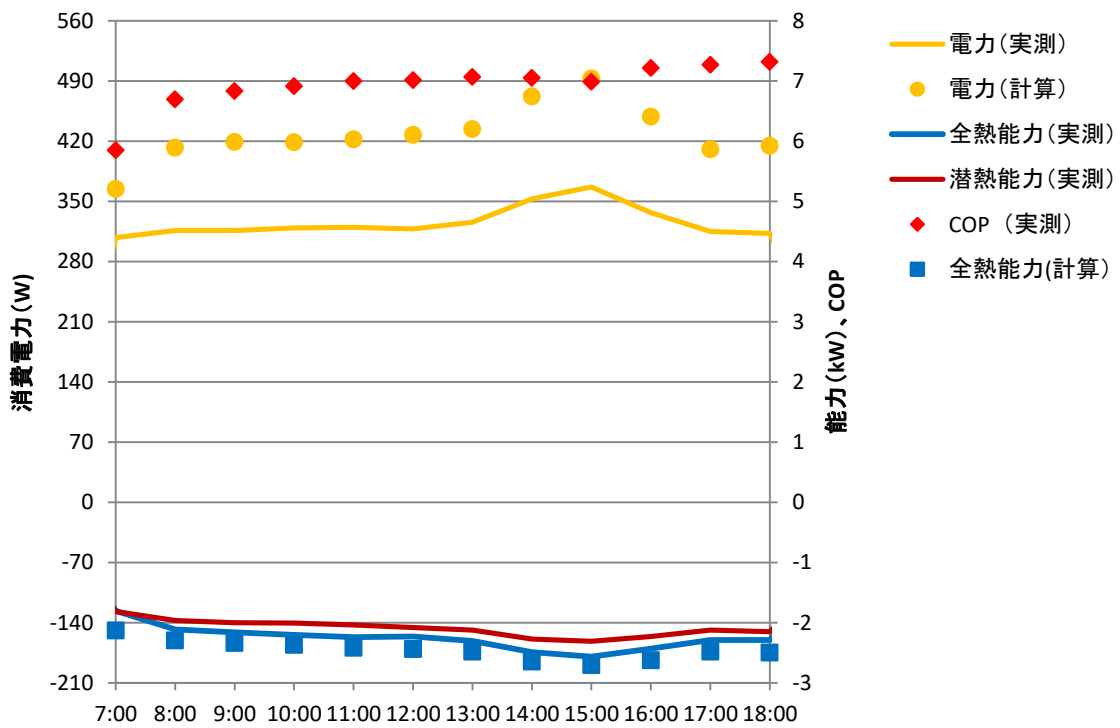


図 117 8/28 運転状態

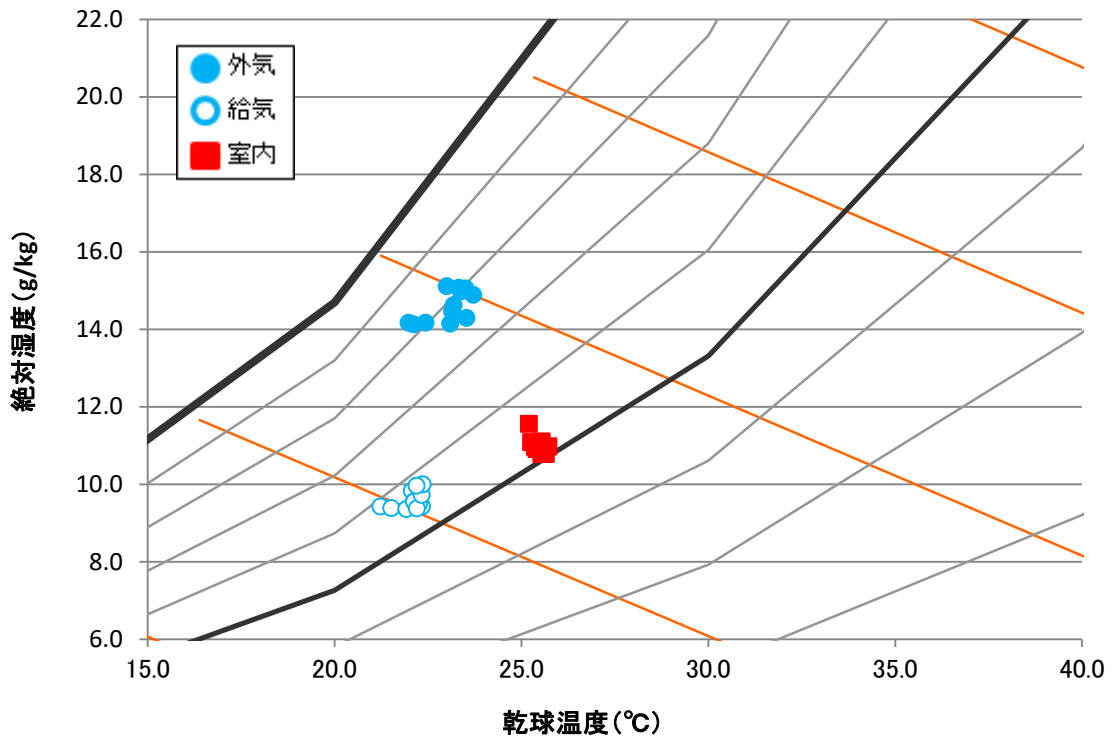


図 118 8/28 空気状態

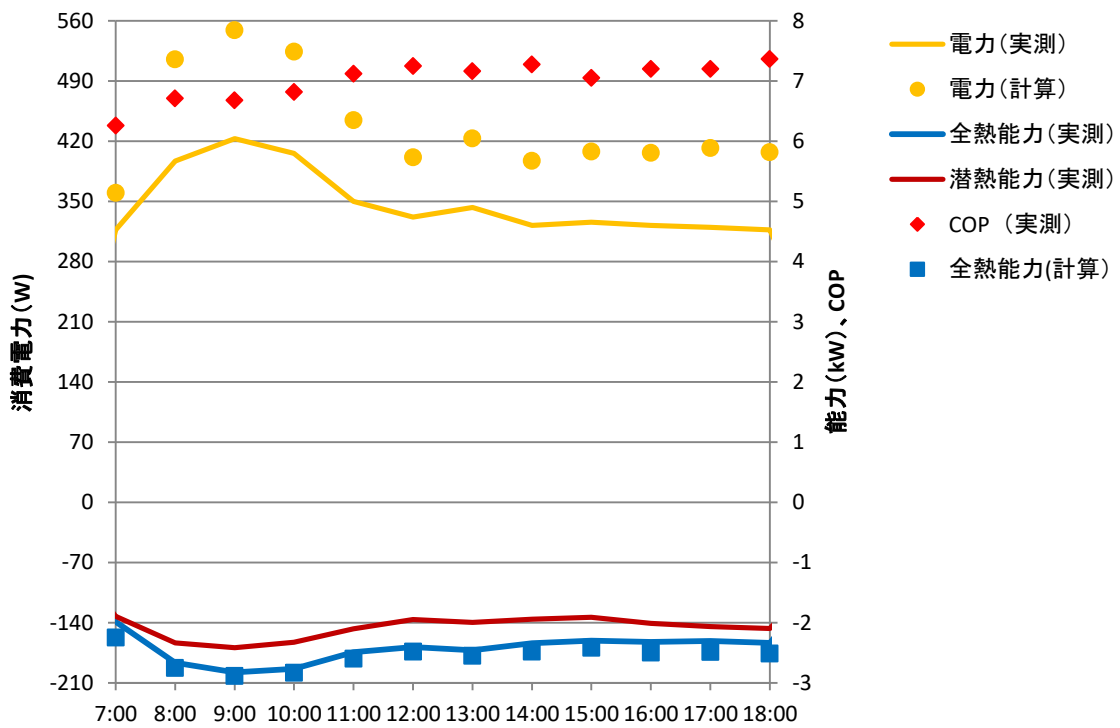


図 119 8/29 運転状態

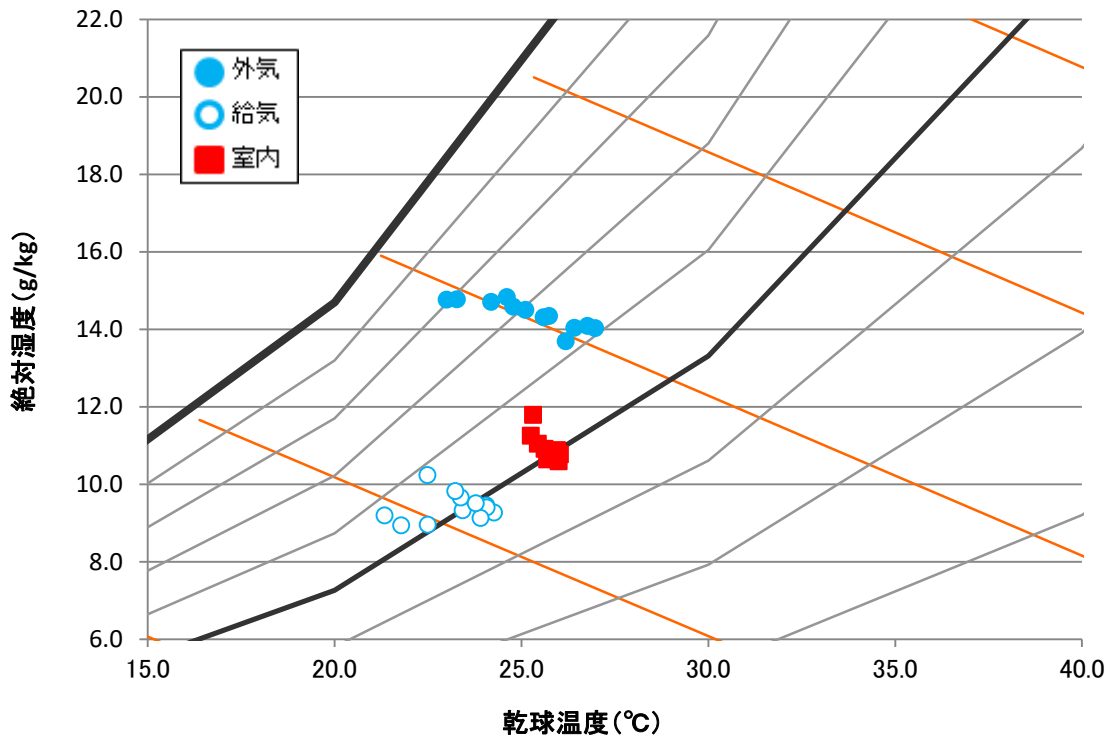


図 120 8/29 空気状態

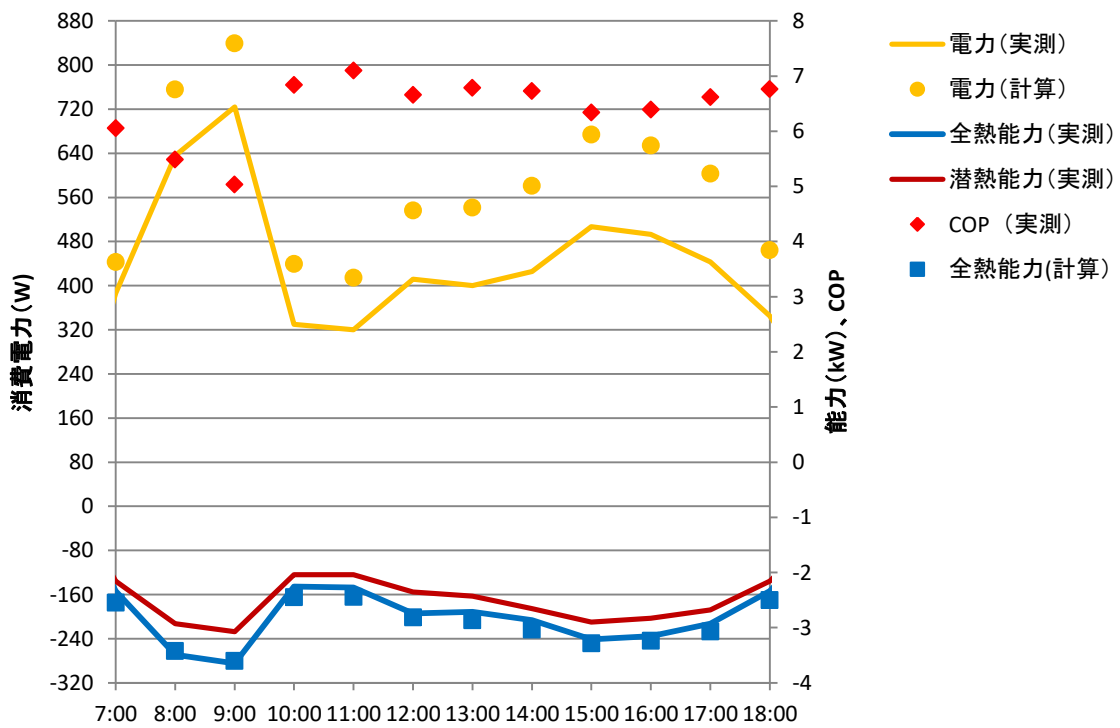


図 121 9/1 運転状態

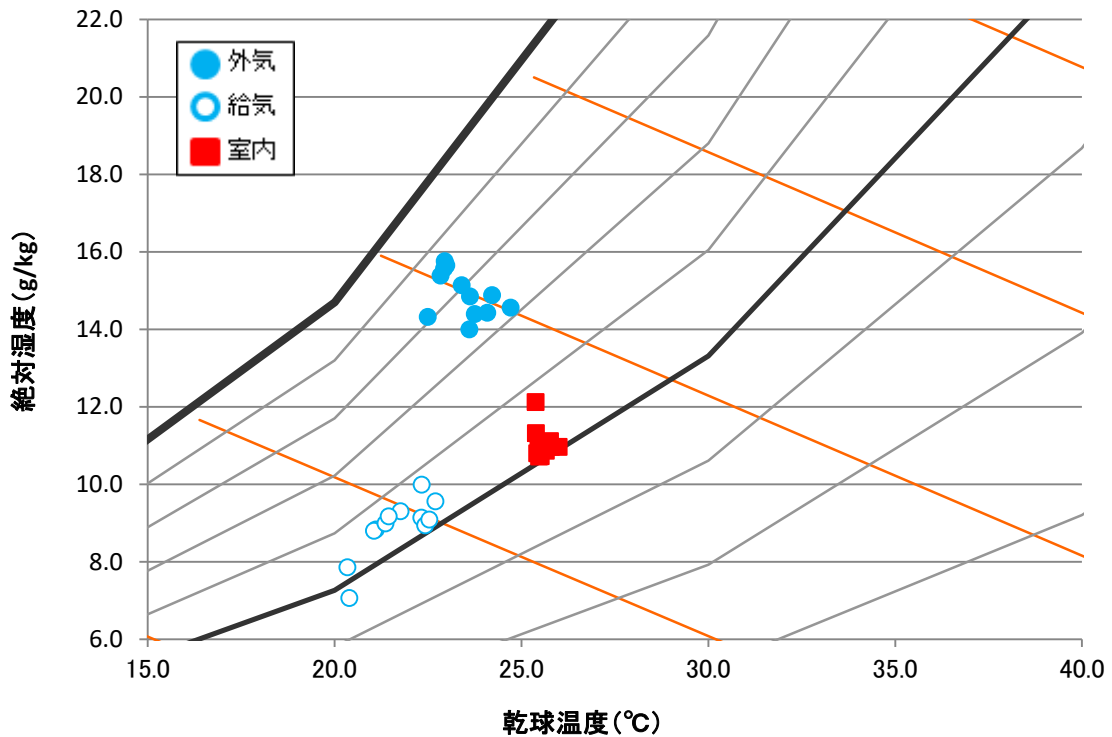


図 122 9/1 空気状態

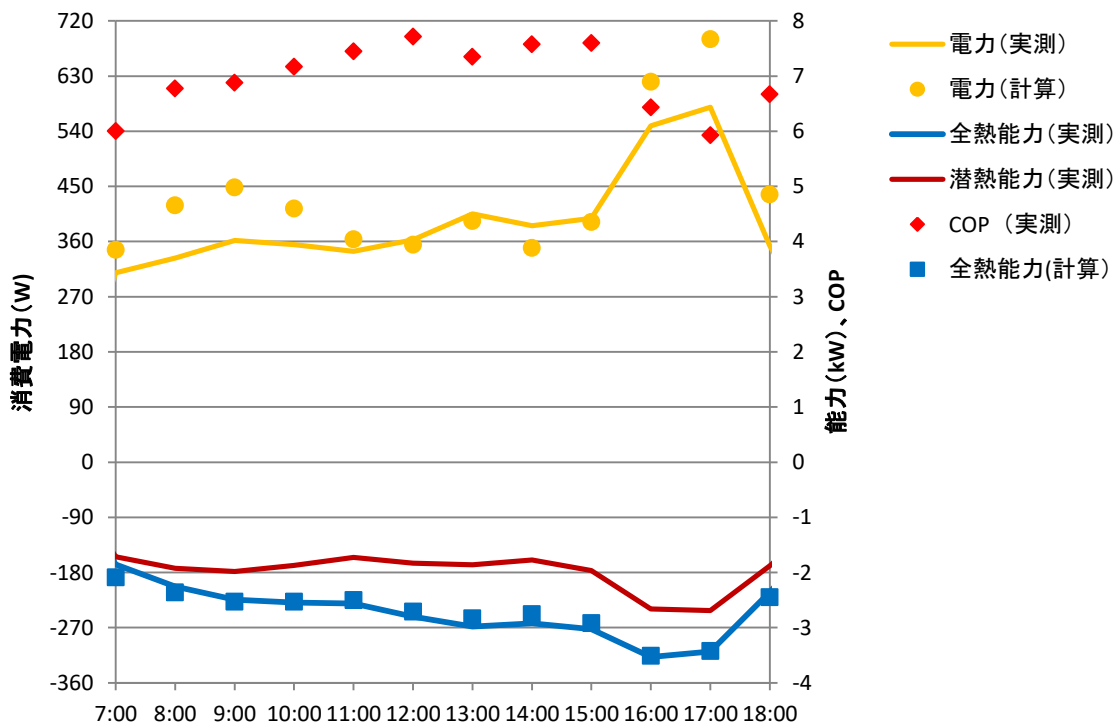


図 123 9/2 運転状態

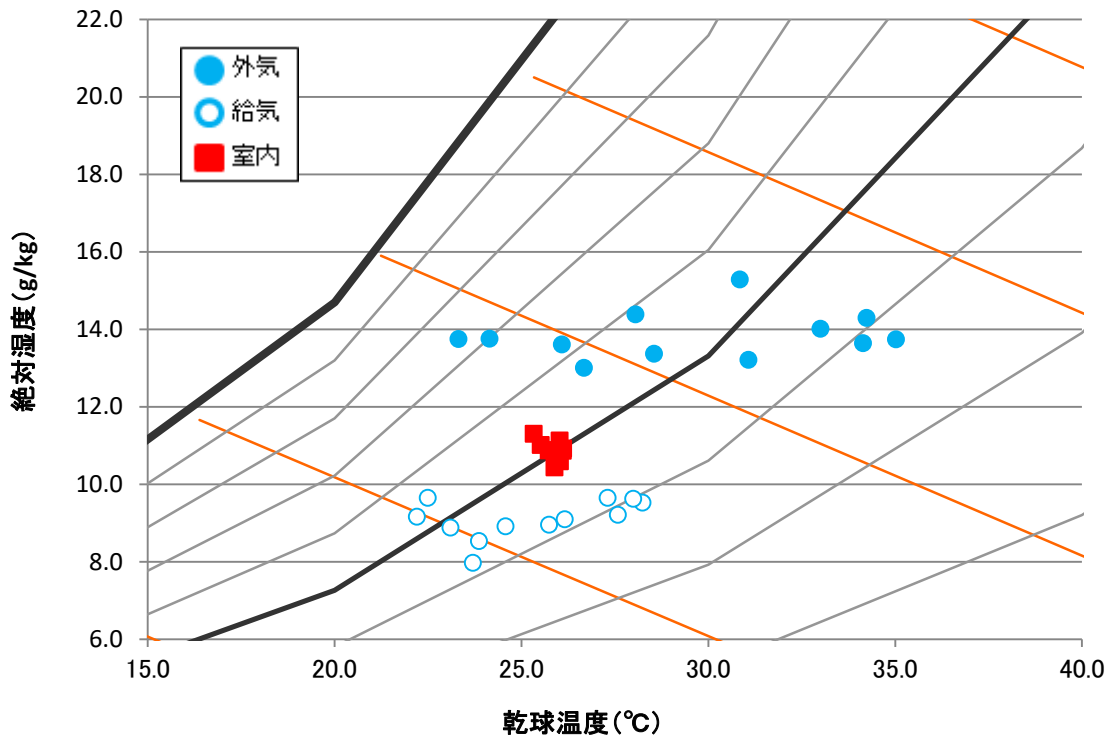


図 124 9/2 空気状態

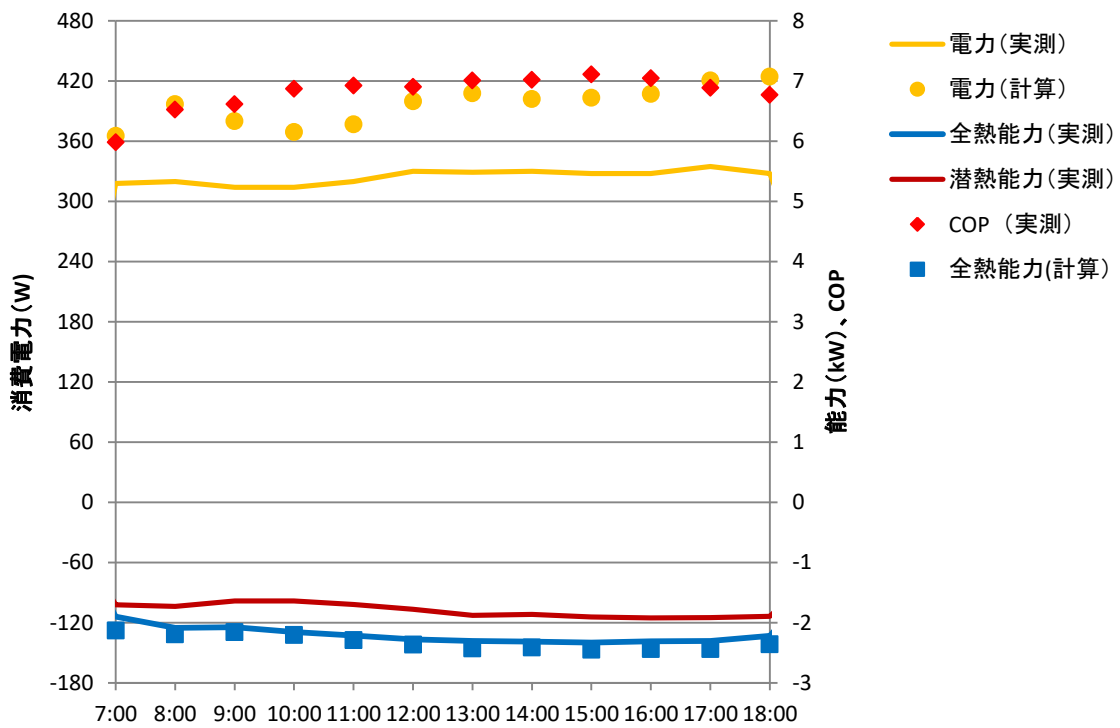


図 125 9/3 運転状態

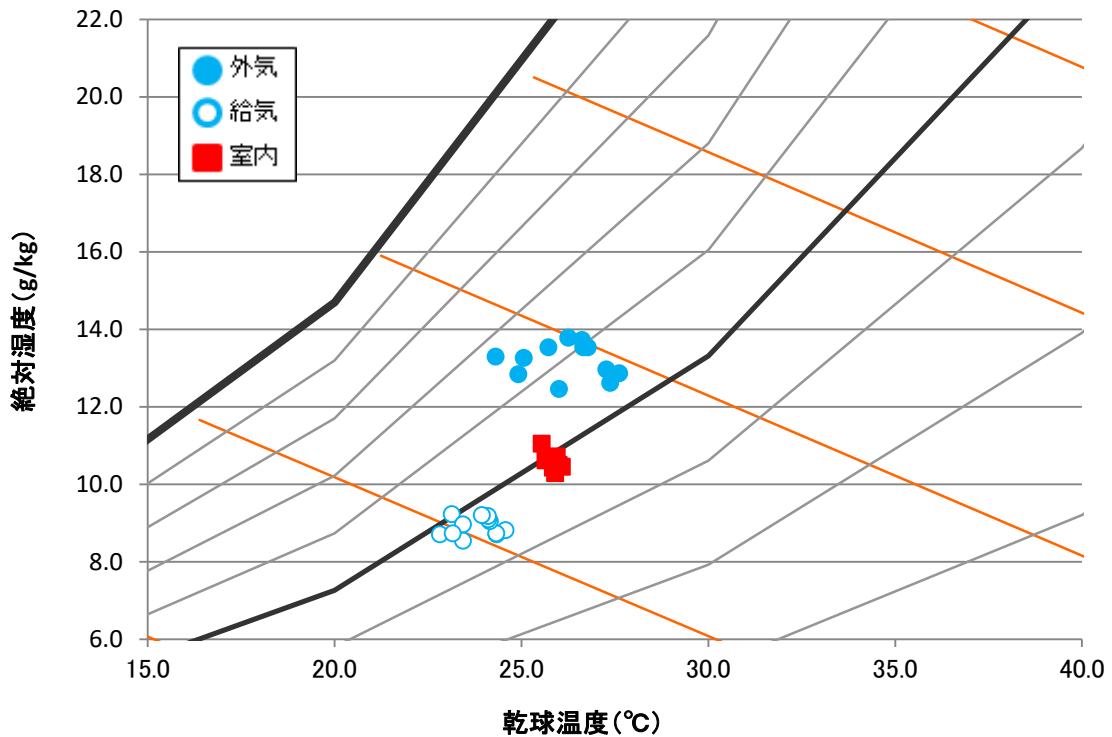


図 126 9/3 空気条件

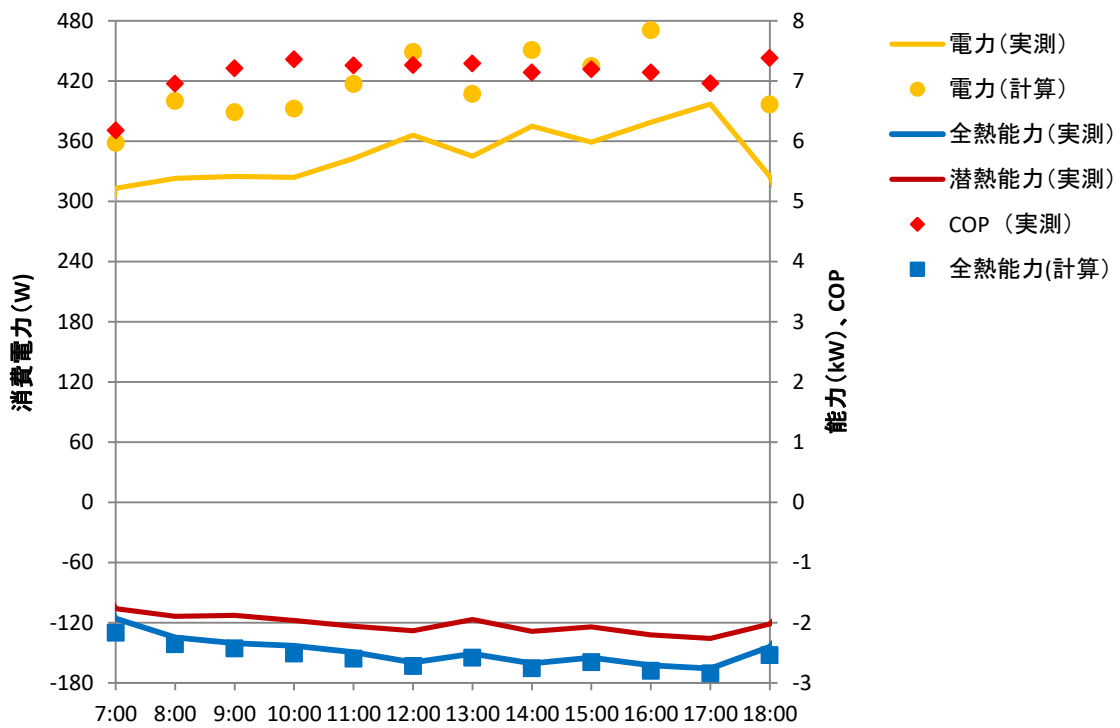


図 127 9/4 運転状態

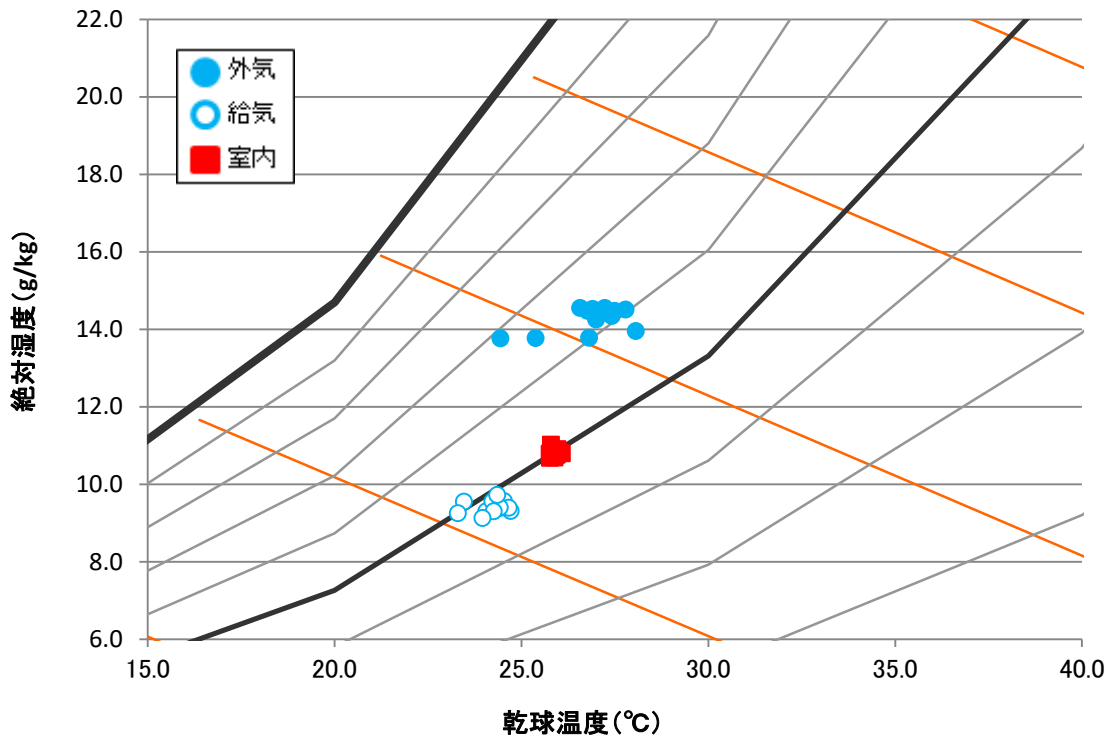


図 128 9/4 空気条件

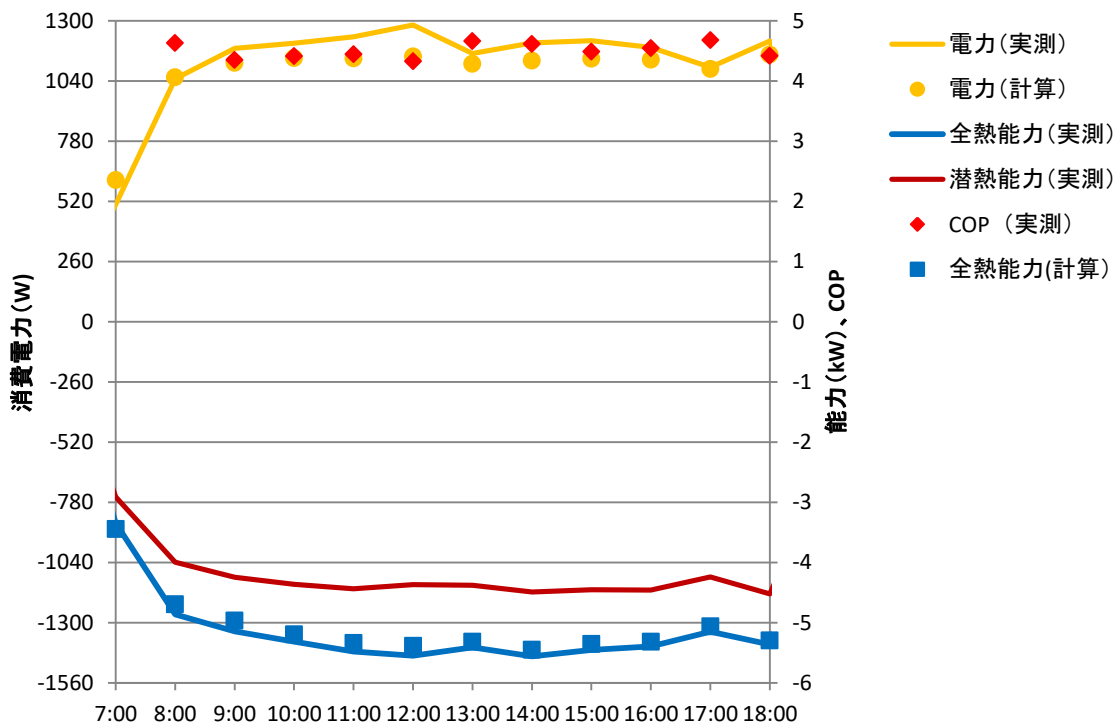


図 129 9/5 運転状態

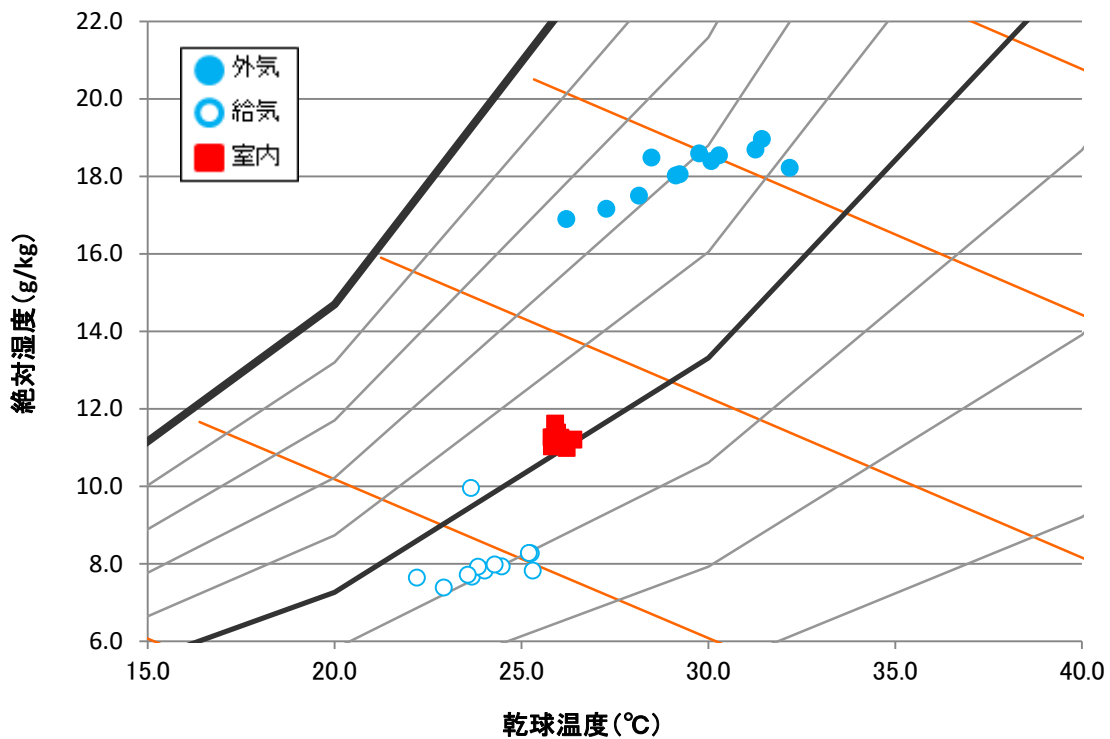


図 130 9/5 空気状態

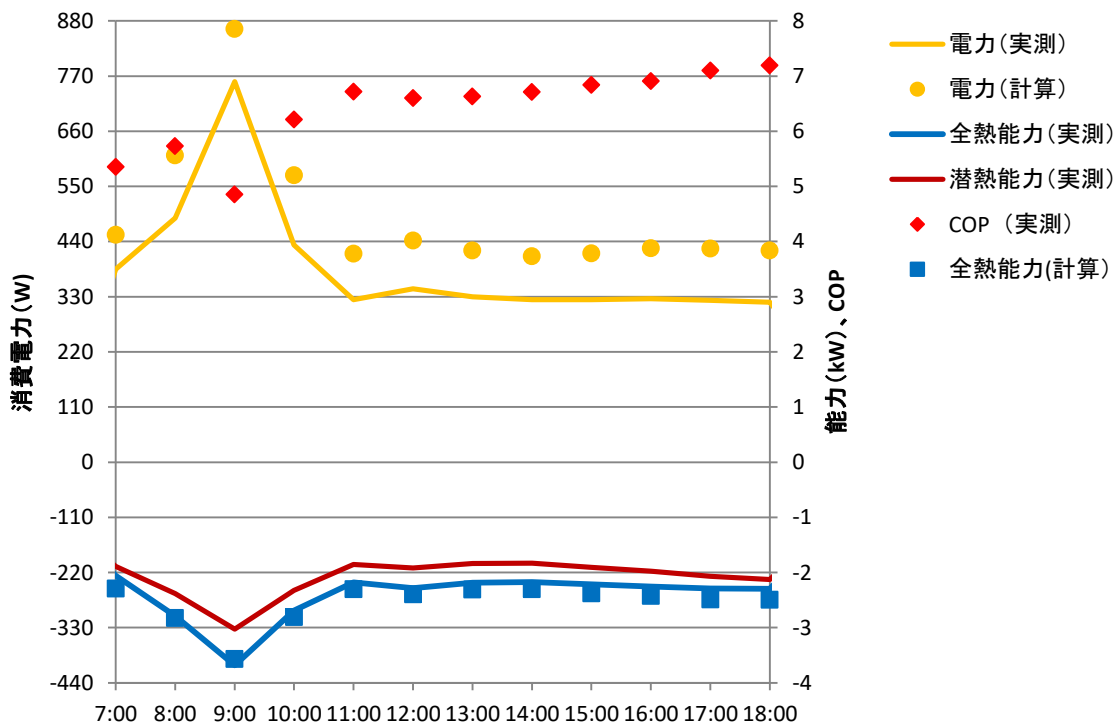


図 131 9/8 運転状態

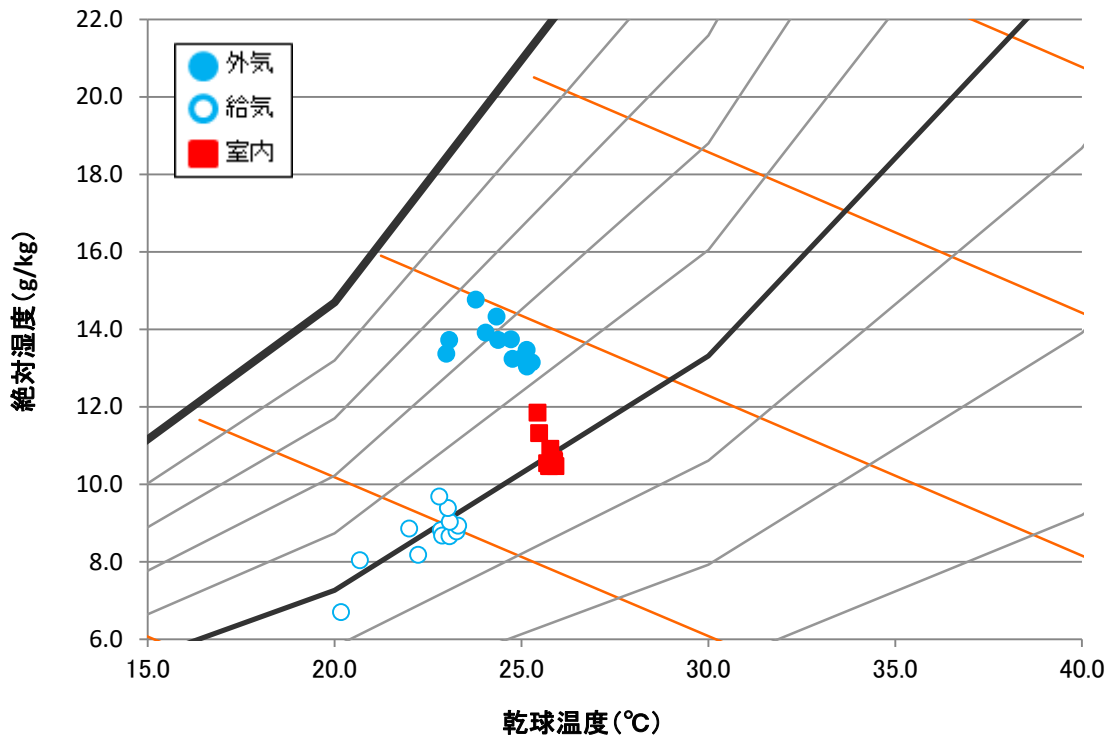


図 132 9/8 空気状態

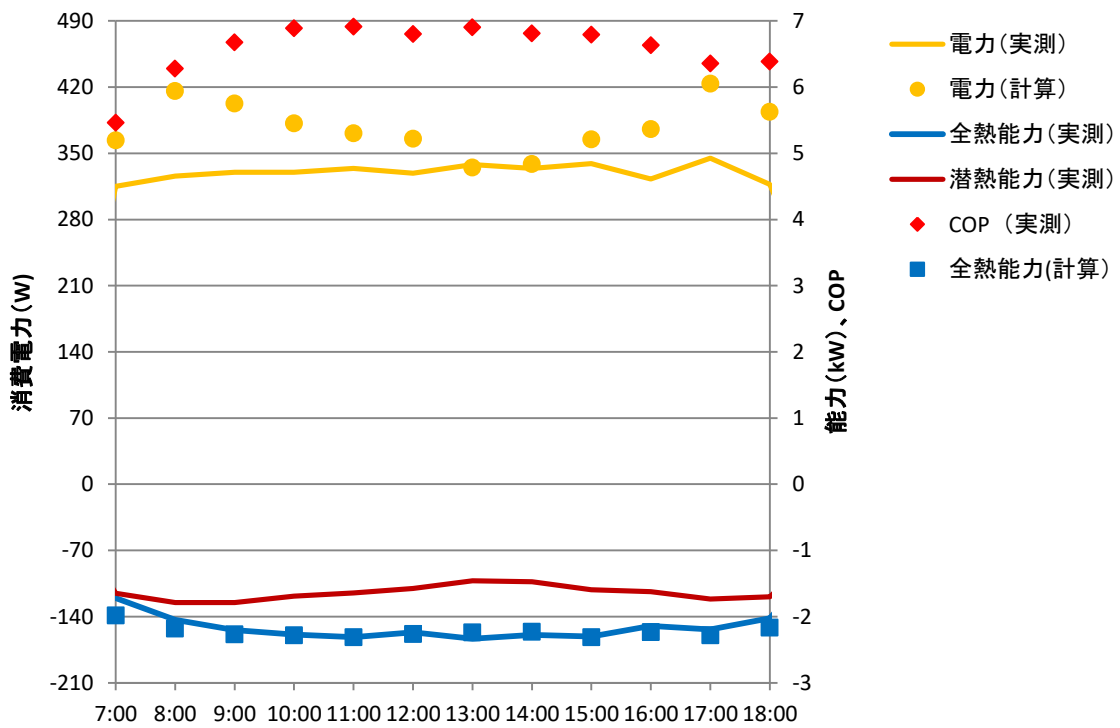


図 133 9/9 運転状態

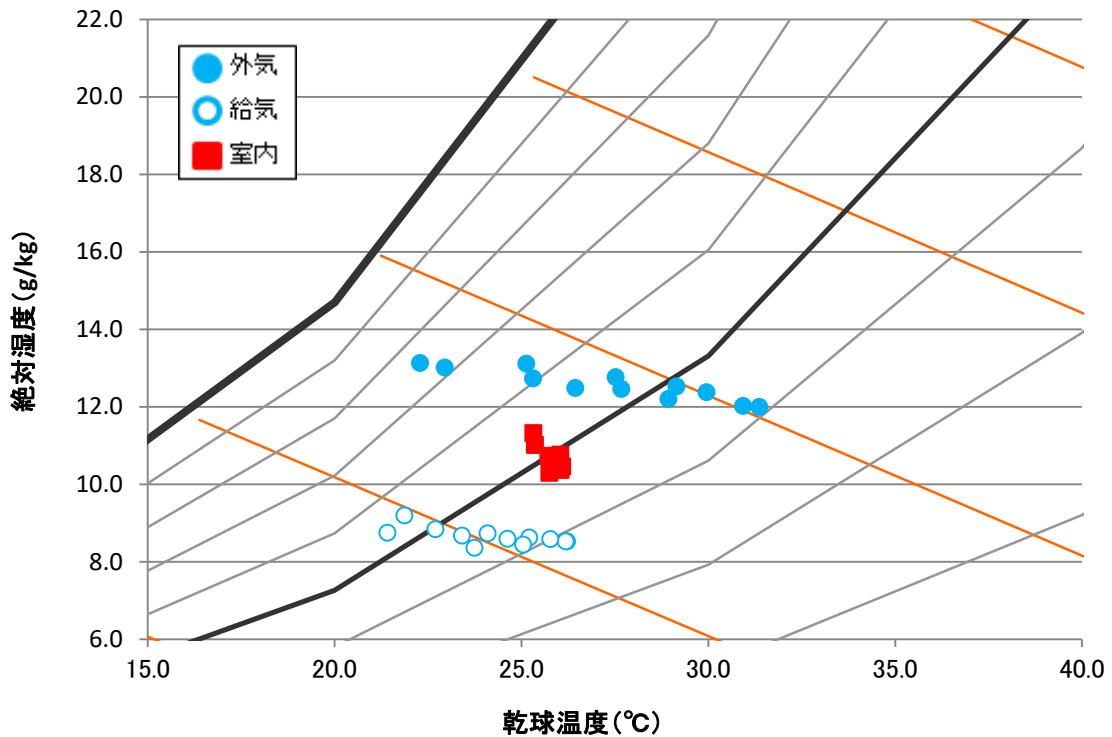


図 134 9/9 空気状態

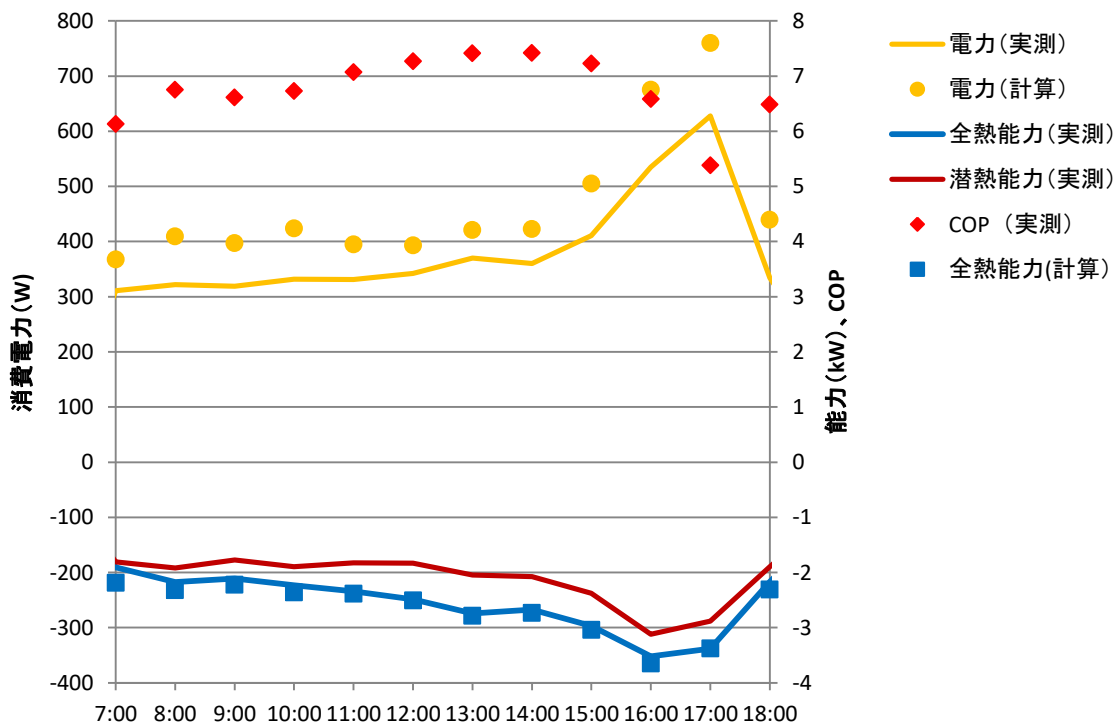


図 135 9/10 運転状態

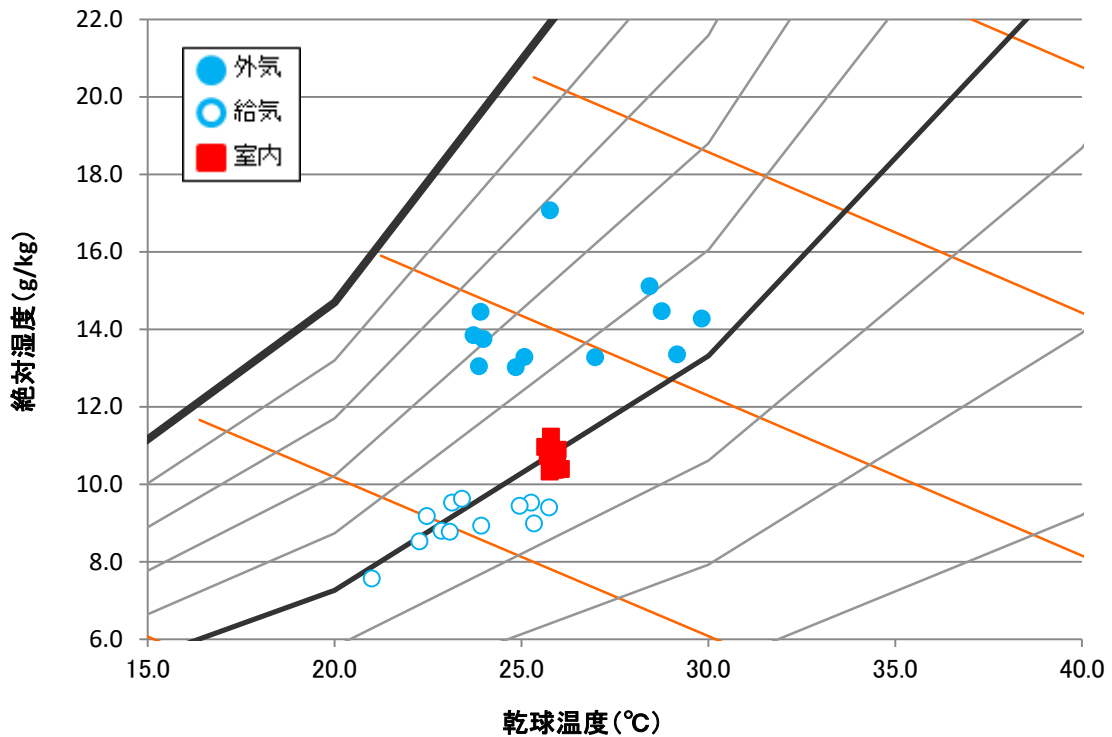


図 136 9/10 空気状態

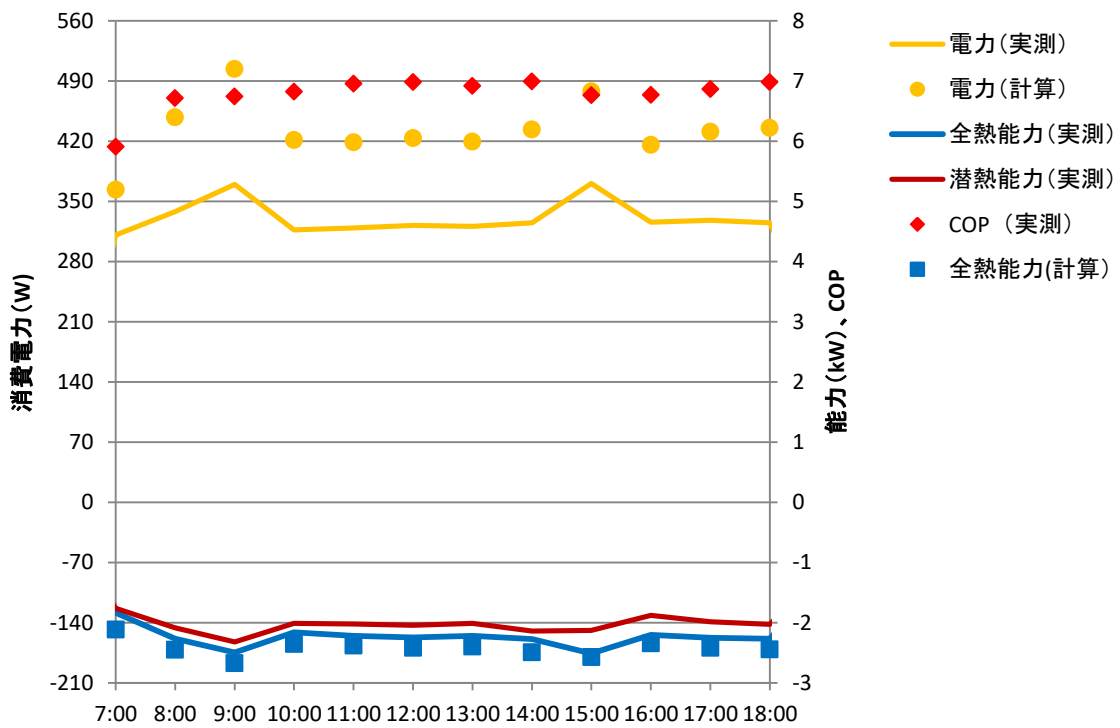


図 137 9/11 運転状態

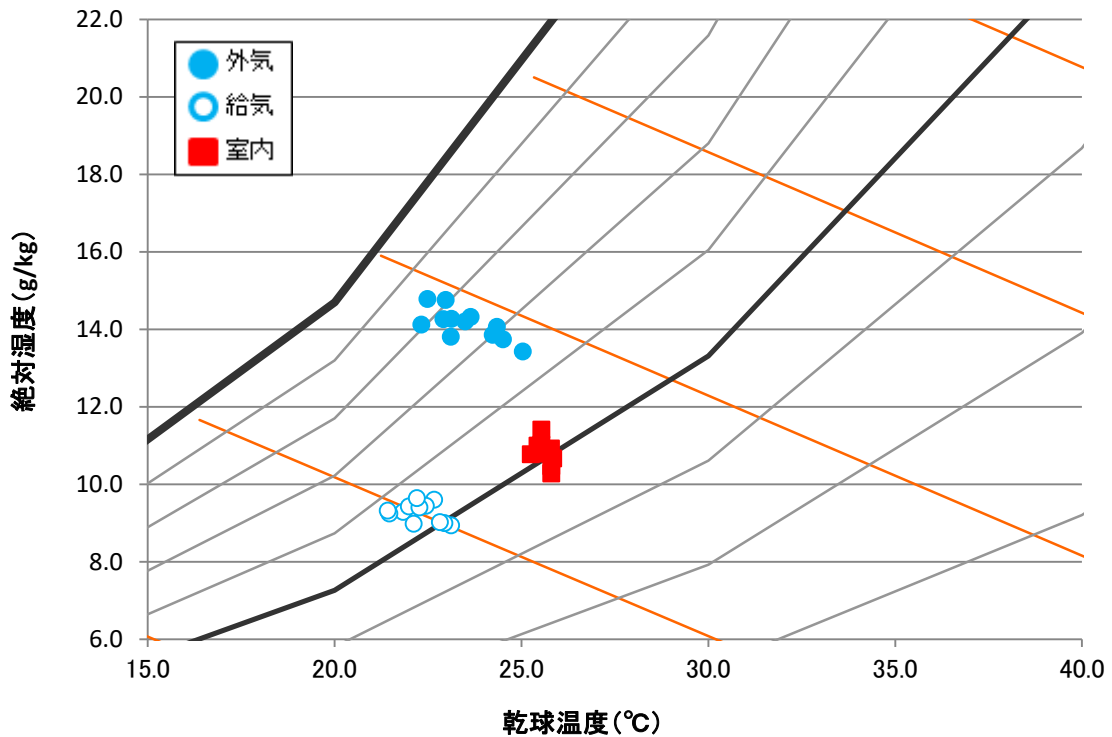


図 138 9/11 空気状態

東京 A ビルにおけるヒートポンプデシカント調湿外調機の性能特性式から計算される消費電力、全熱能力、潜熱能力と実測値の比較データ（冬季）

データは、日毎の機器の運転状況（消費電力量（実測値、特性式算出値）、全熱能力（実測値、特性式算出値）、潜熱能力、COP）の 1 時間毎の平均値および室内、室外、給気の空気状態の 1 時間毎の平均値を示す。

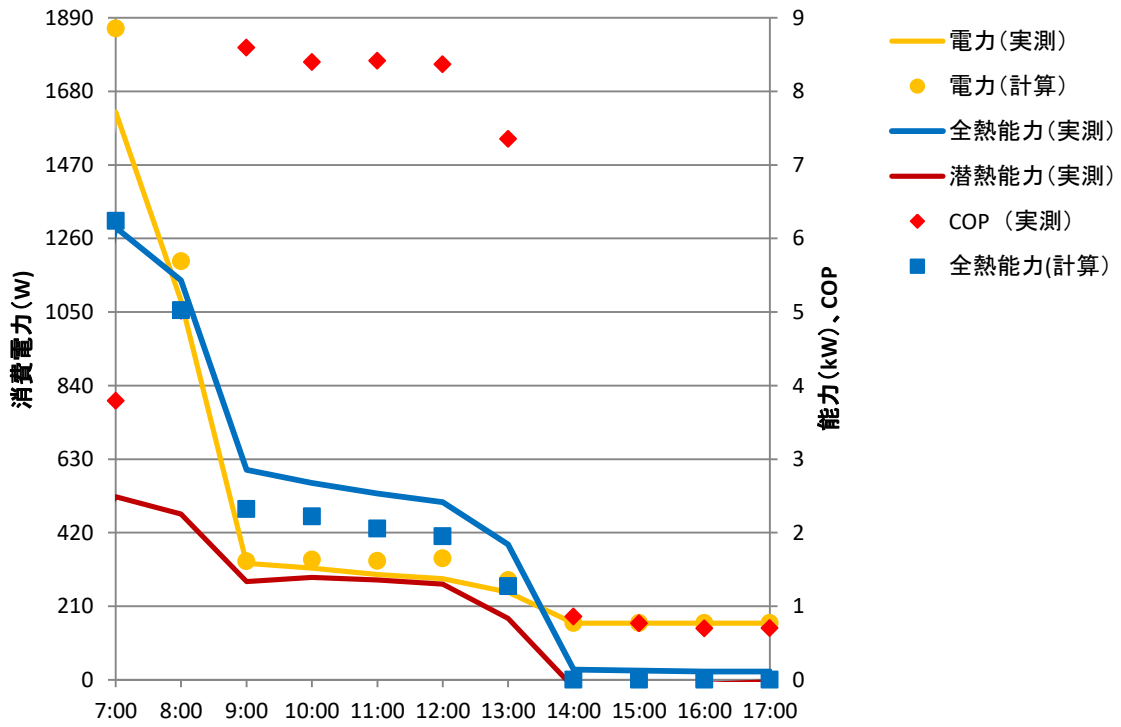


図 139 1/30 運転状況

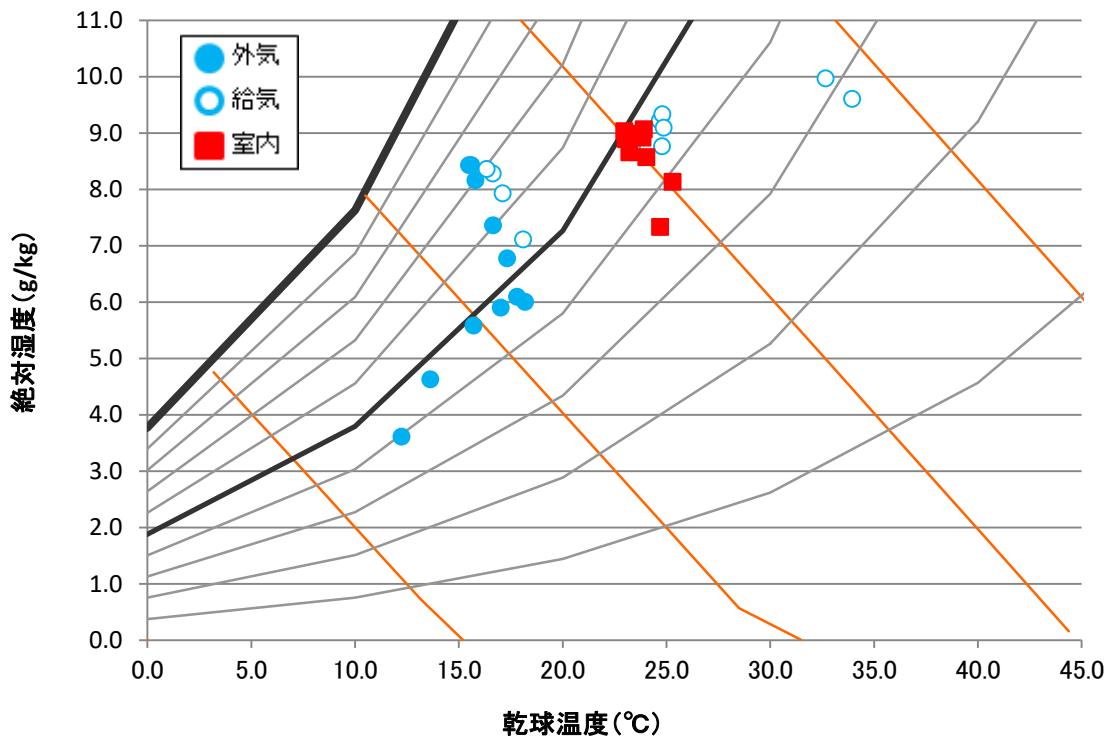


図 140 1/30 空気状況

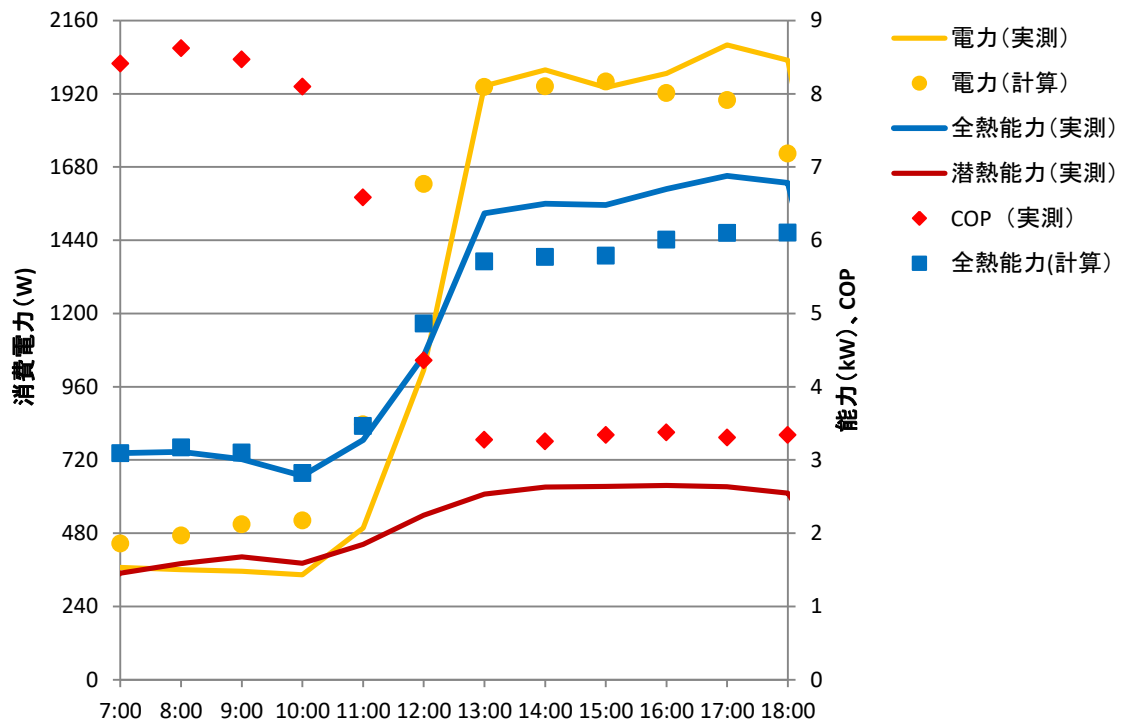


図 141 1/31 運転状況

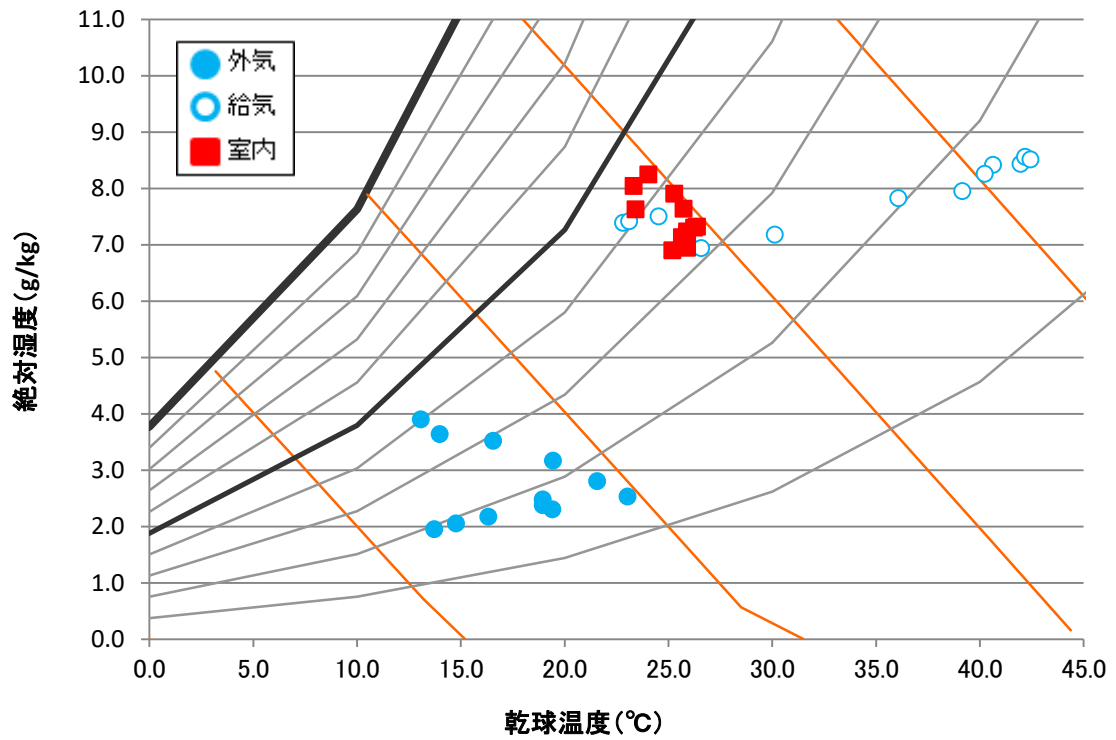


図 142 1/31 空気状況

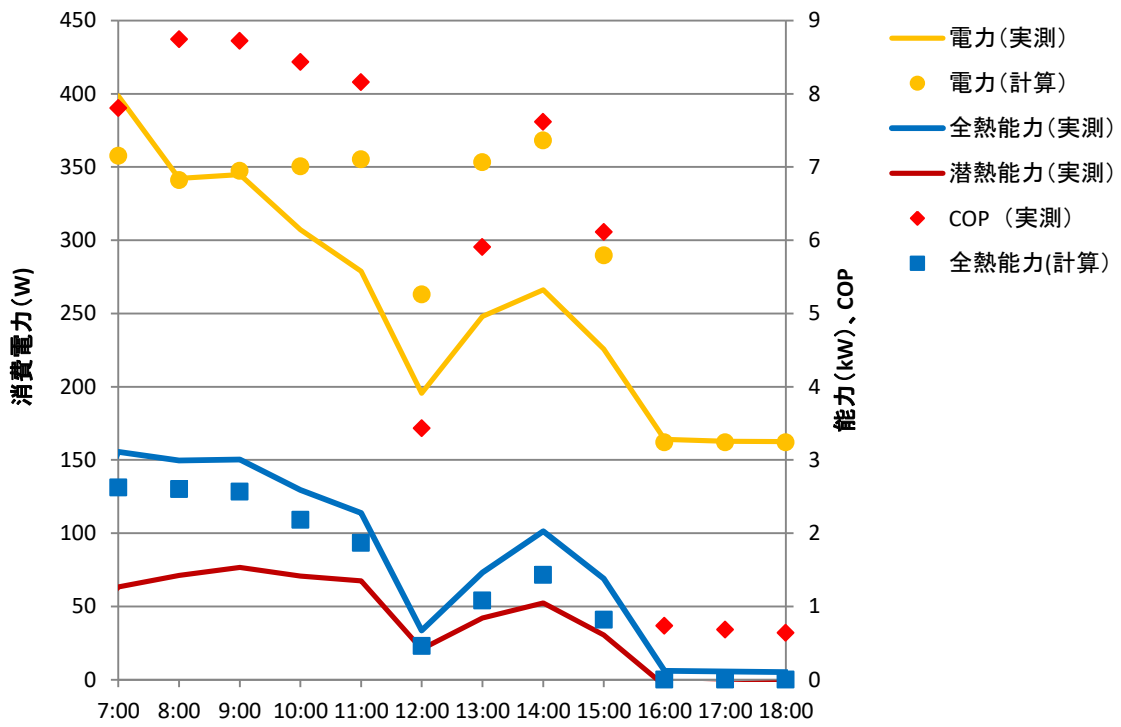


図 143 2/3 運転状態

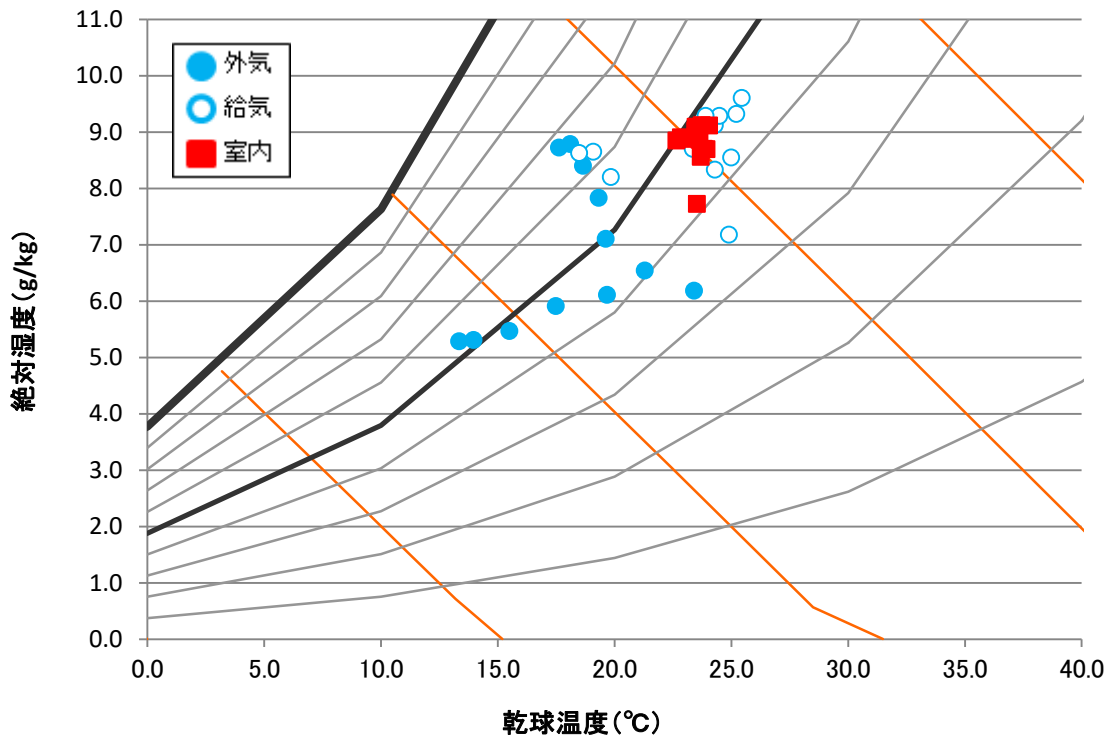


図 144 2/3 空気状態

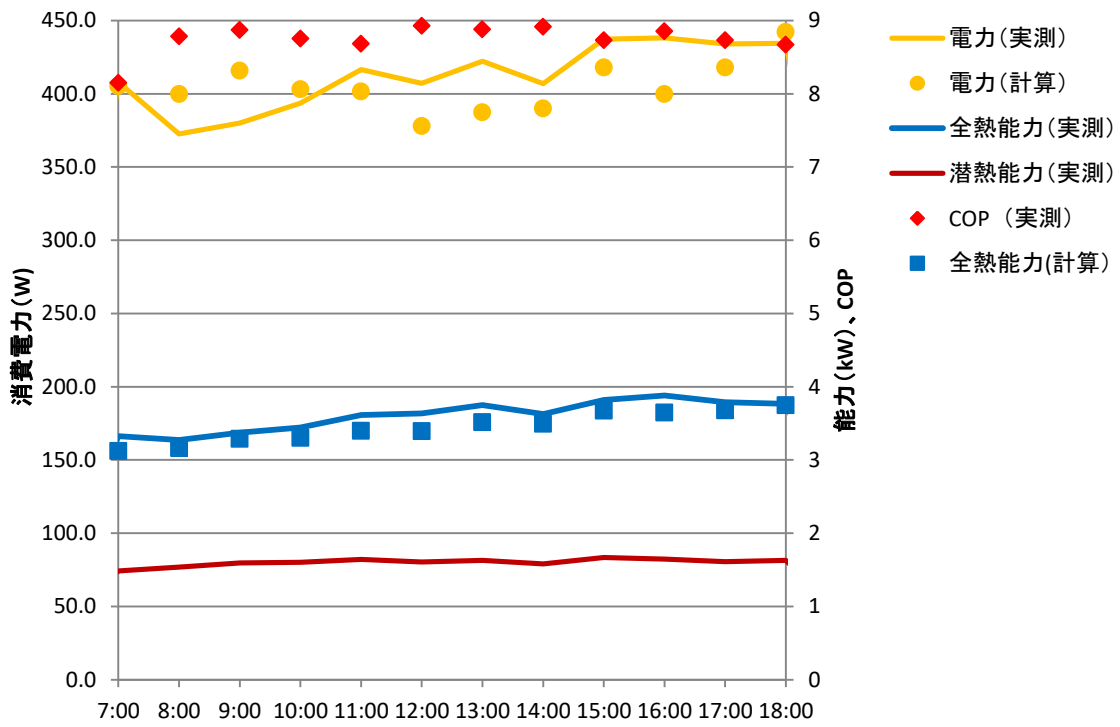


図 145 2/4 運転状態

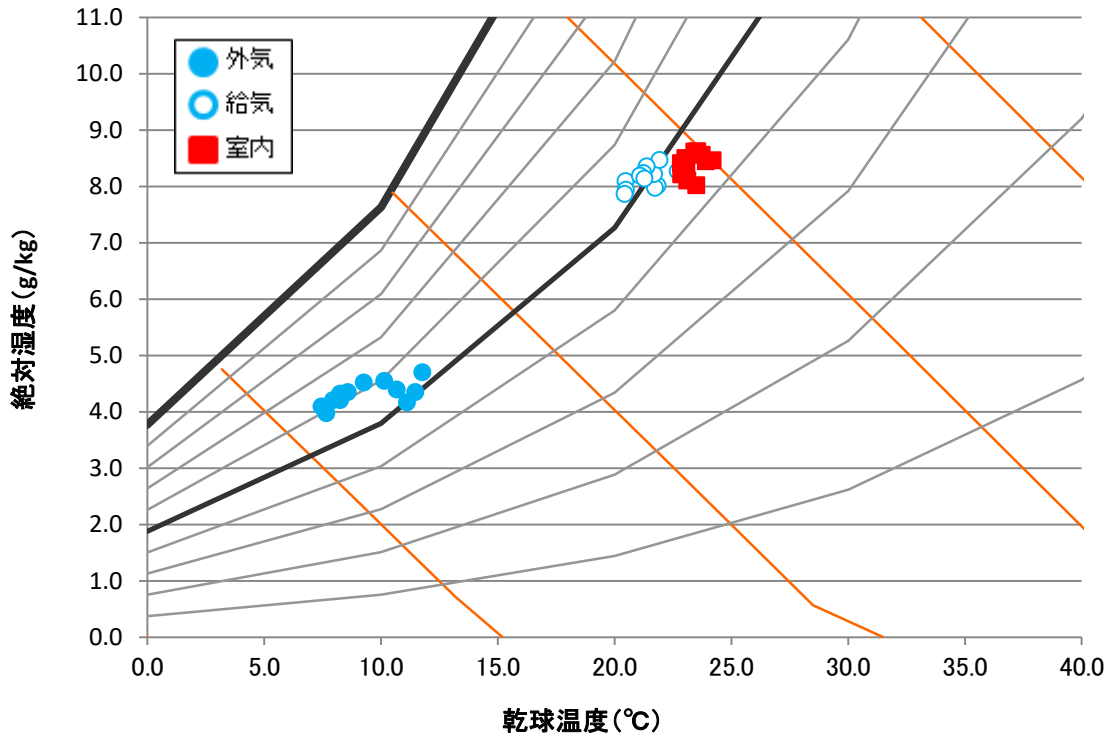


図 146 2/4 空気状態

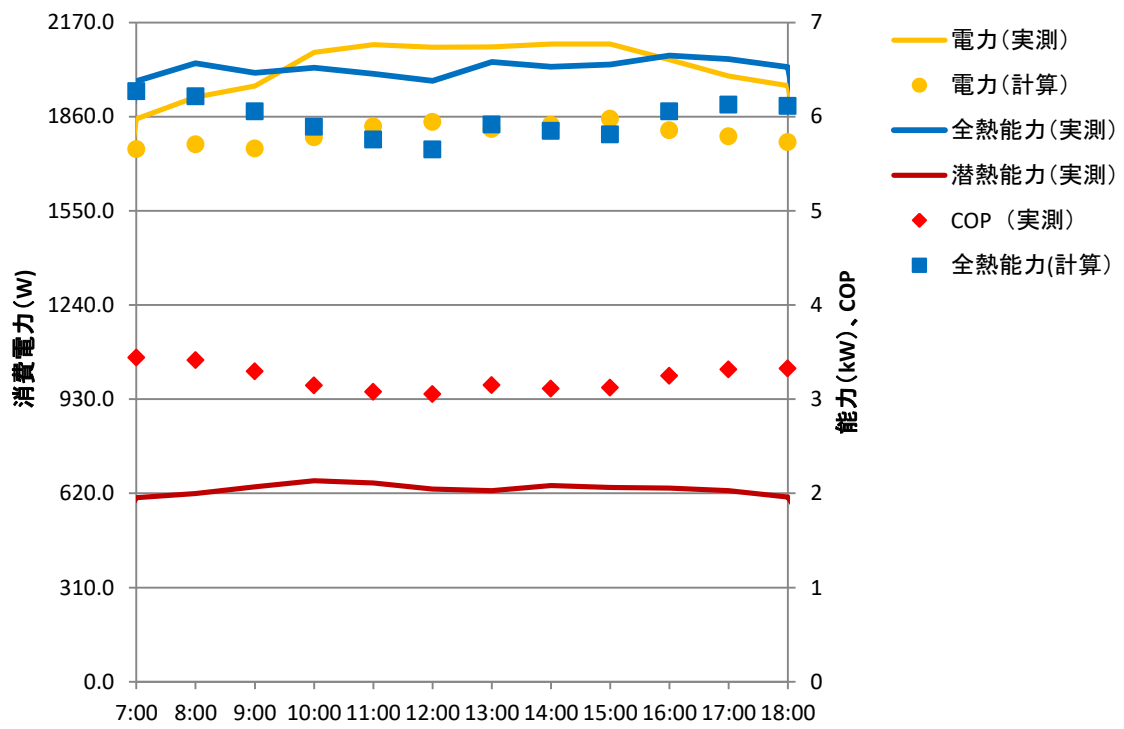


図 147 2/5 運転状態

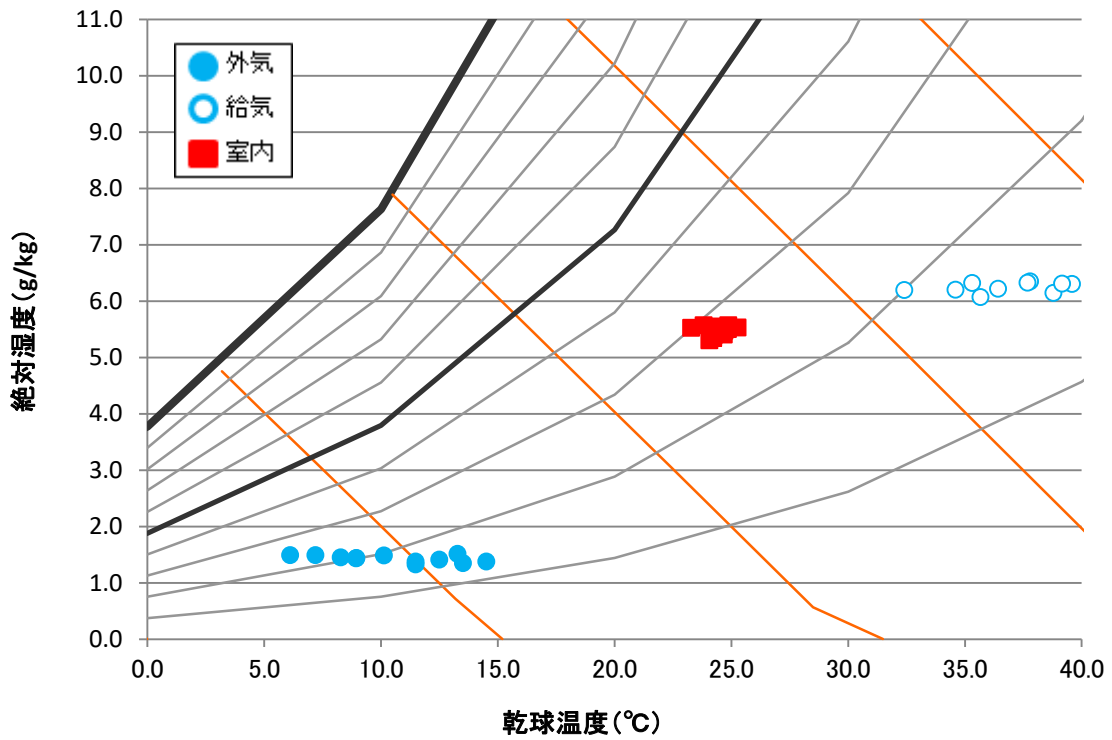


図 148 2/5 空気状態

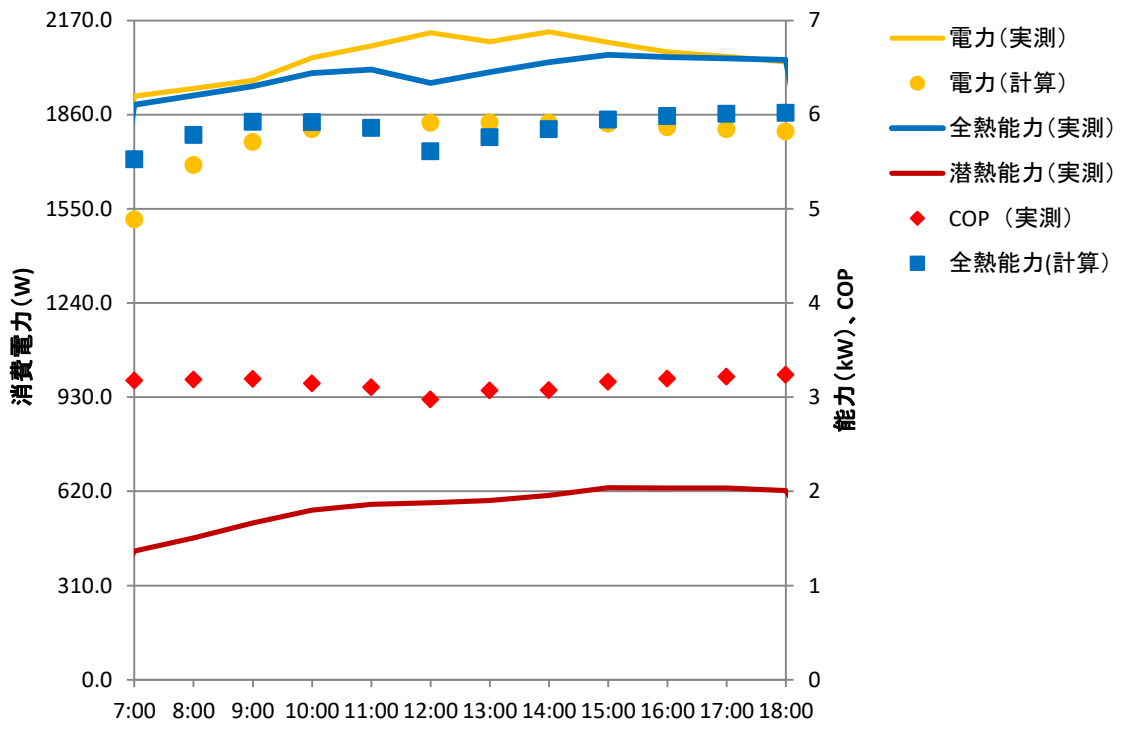


図 149 2/6 運転状態

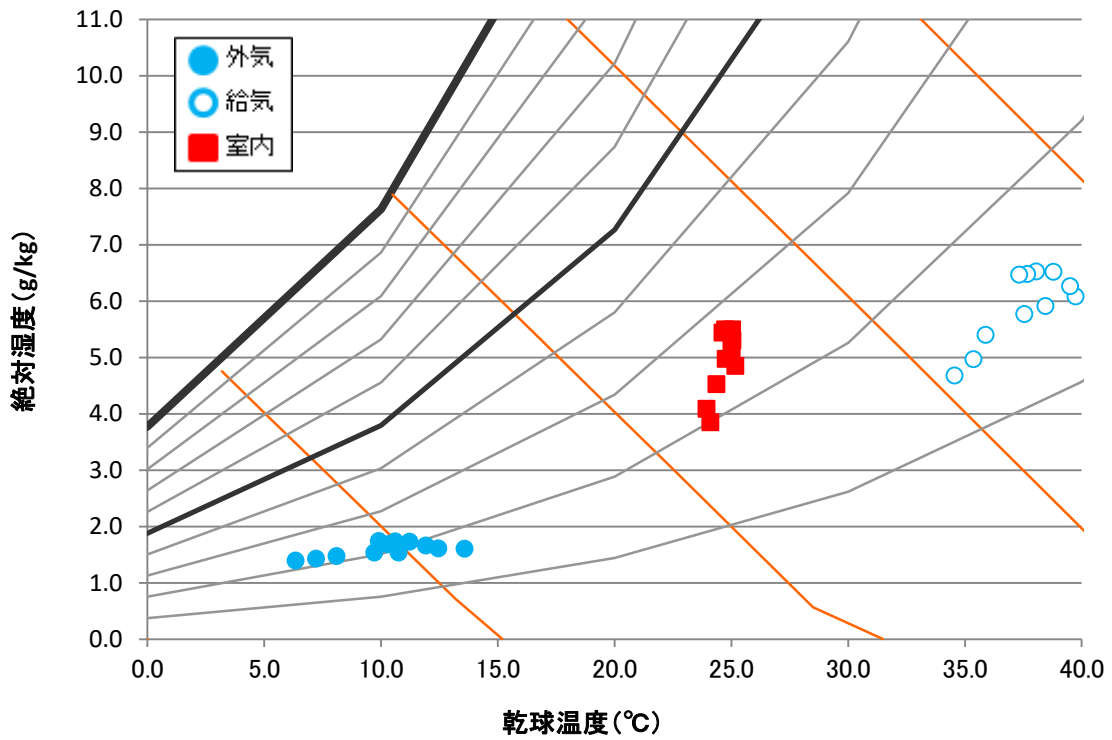


図 150 2/6 空気状態

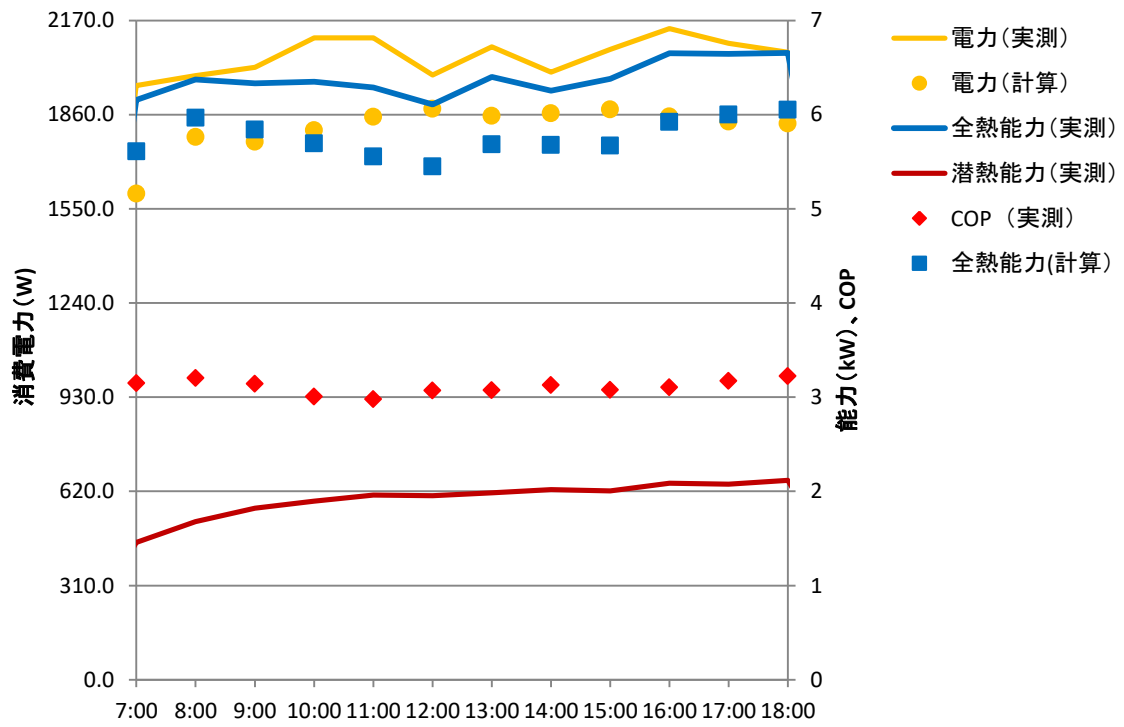


図 151 2/7 運転状態

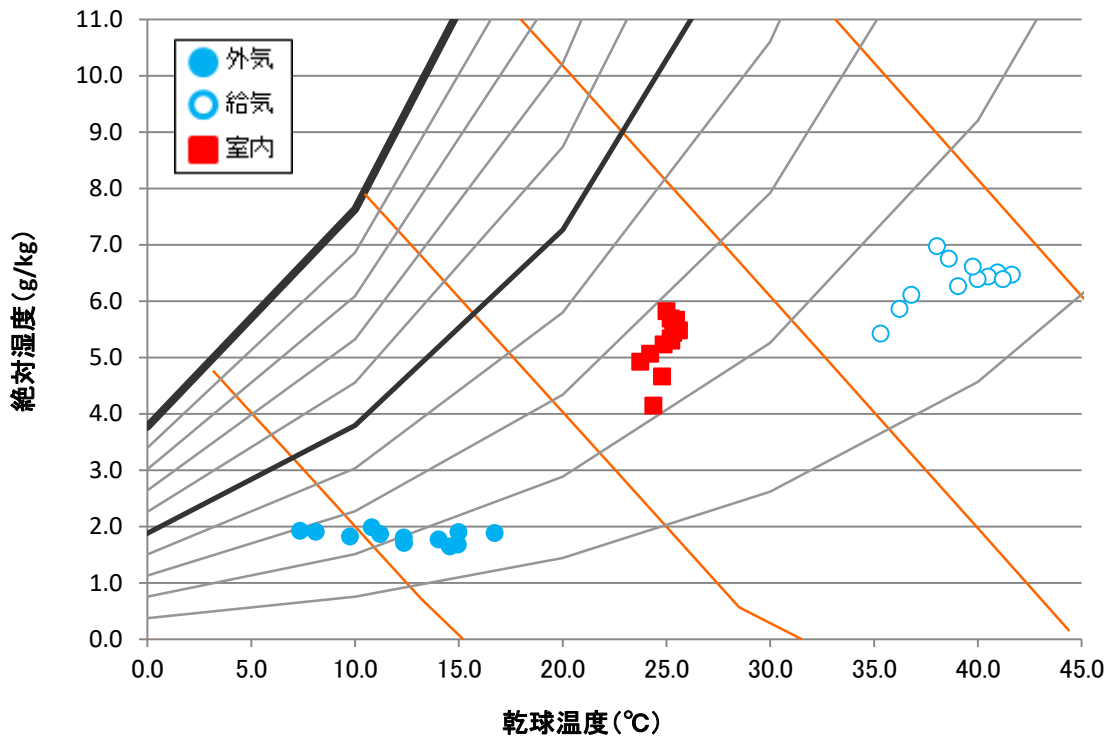


図 152 2/7 空気状態

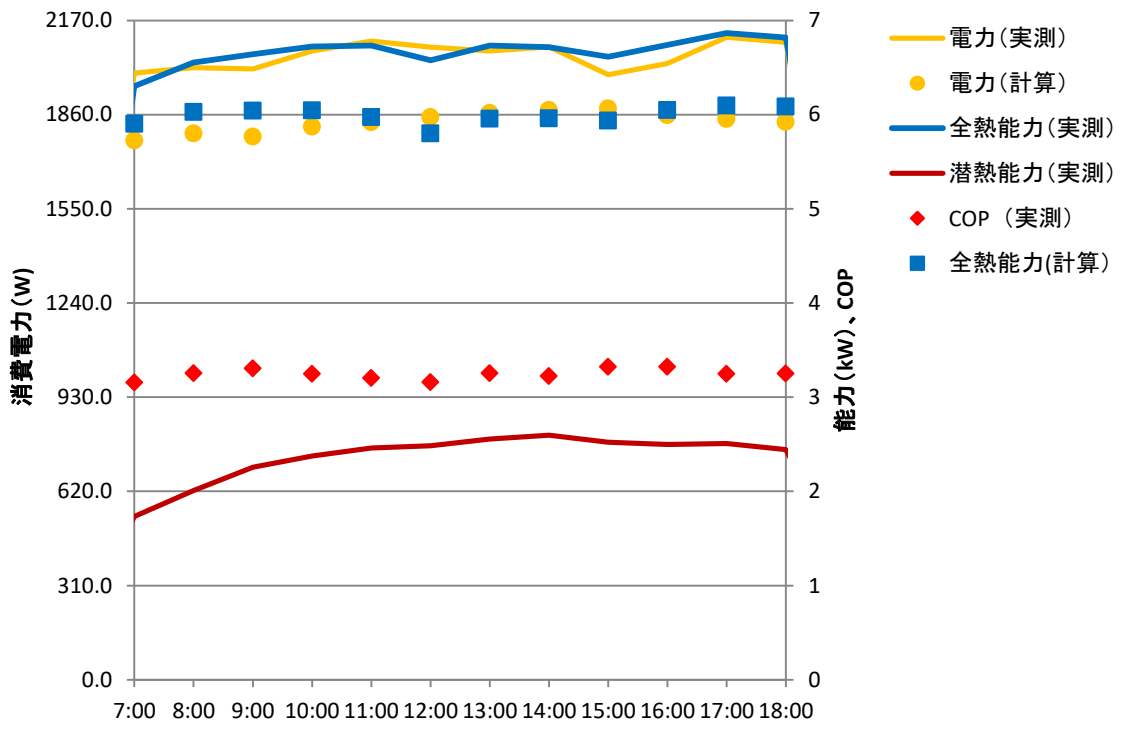


図 153 2/10 運転状態

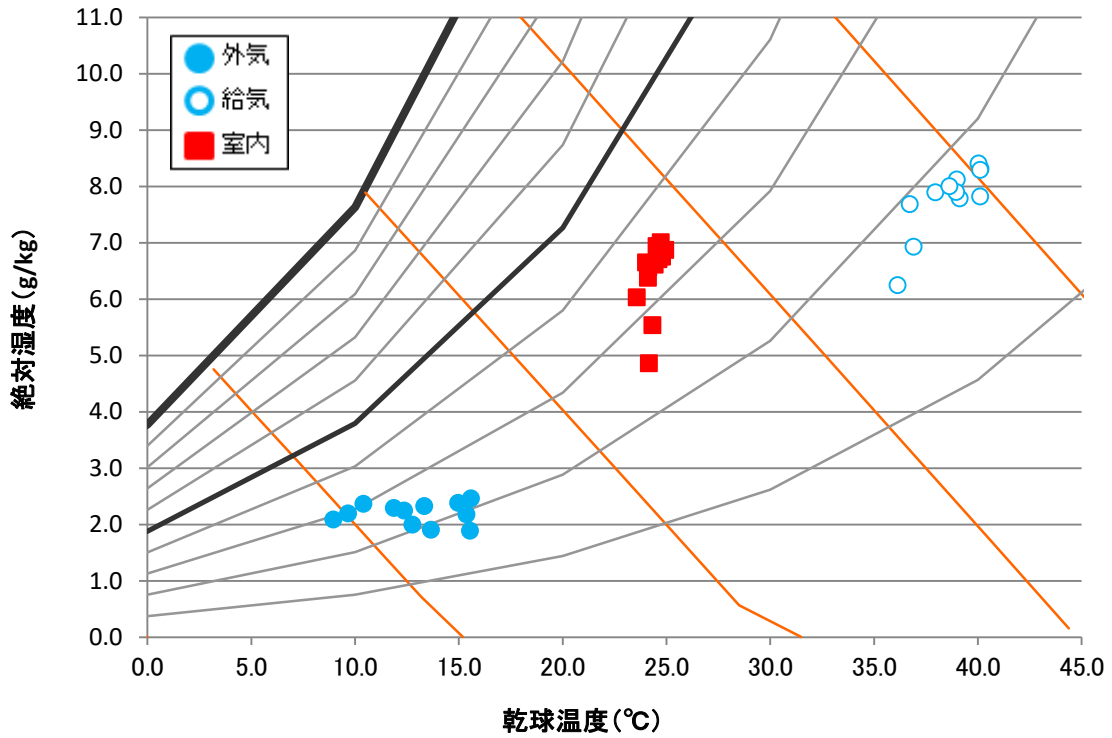


図 154 2/10 空気状態

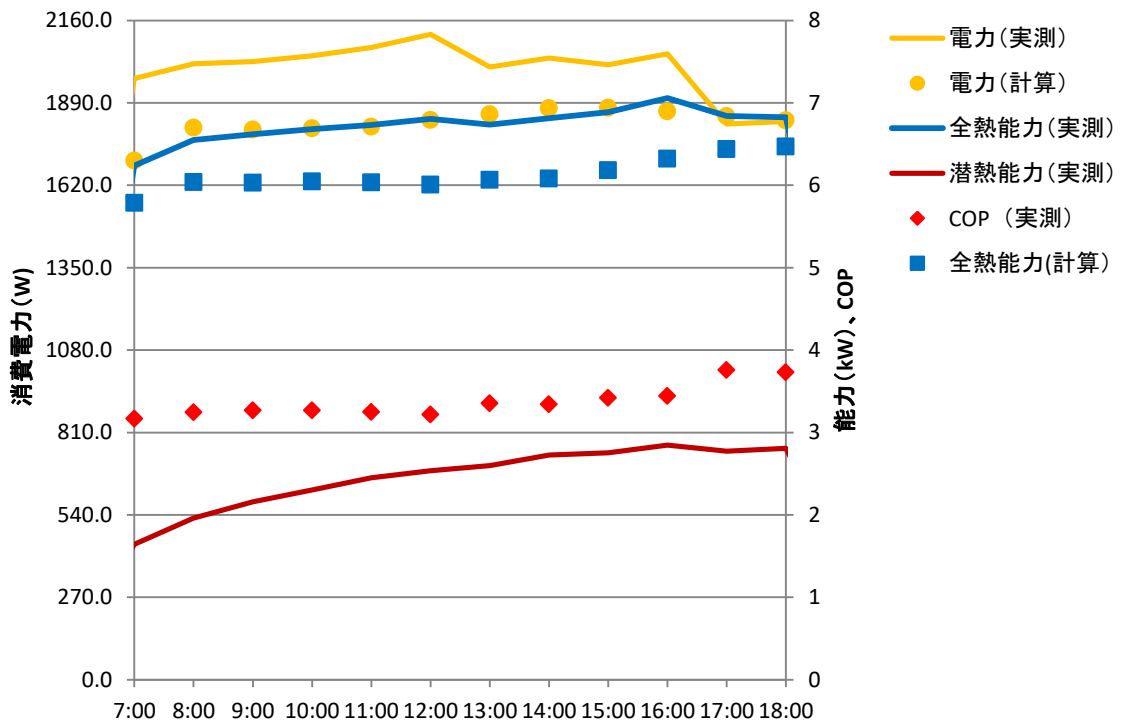


図 155 2/12 運転状態

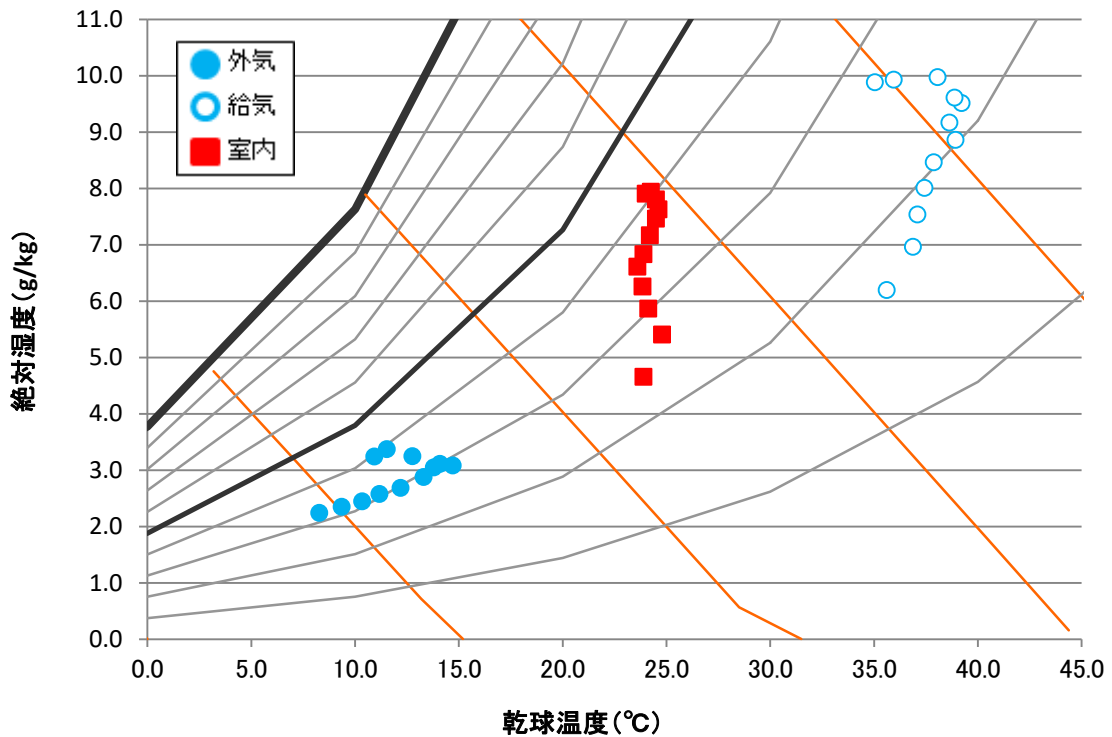


図 156 2/12 空気状態

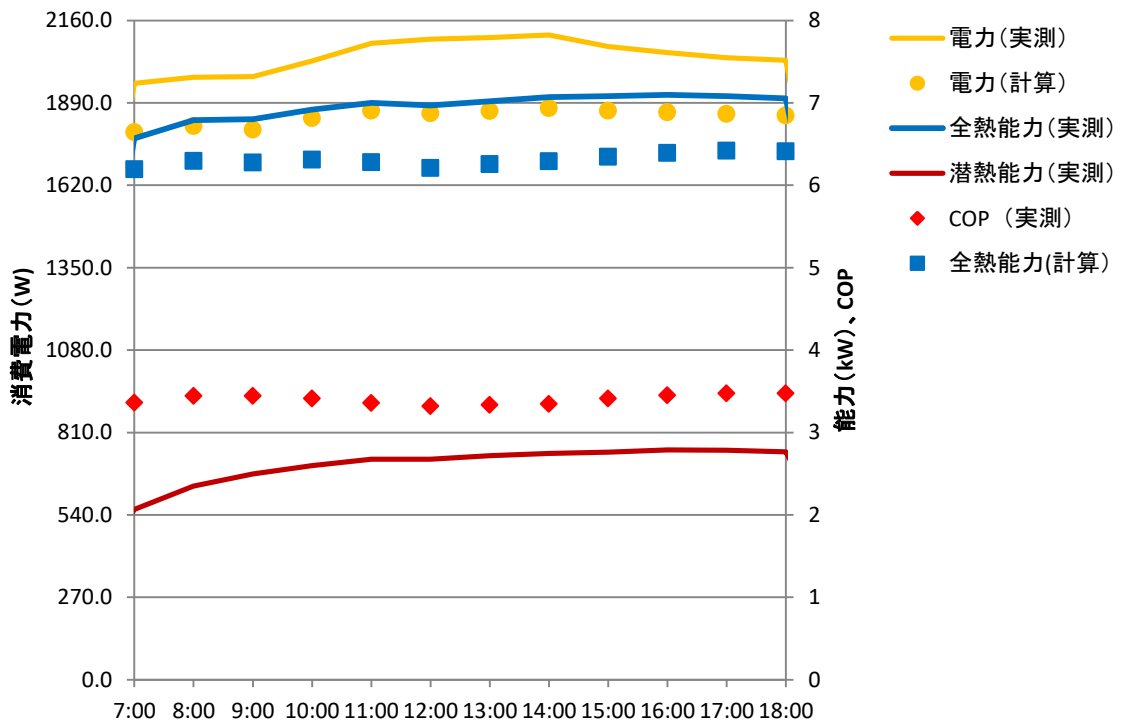


図 157 2/13 運転状態

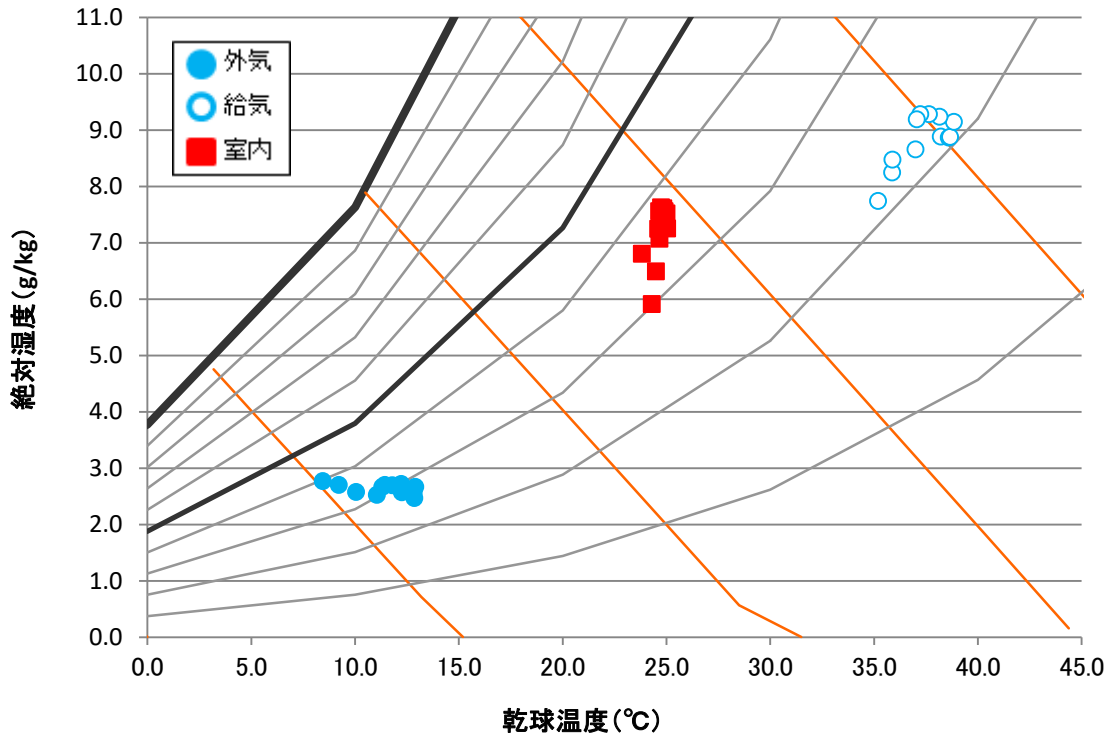


図 158 2/13 運転状態

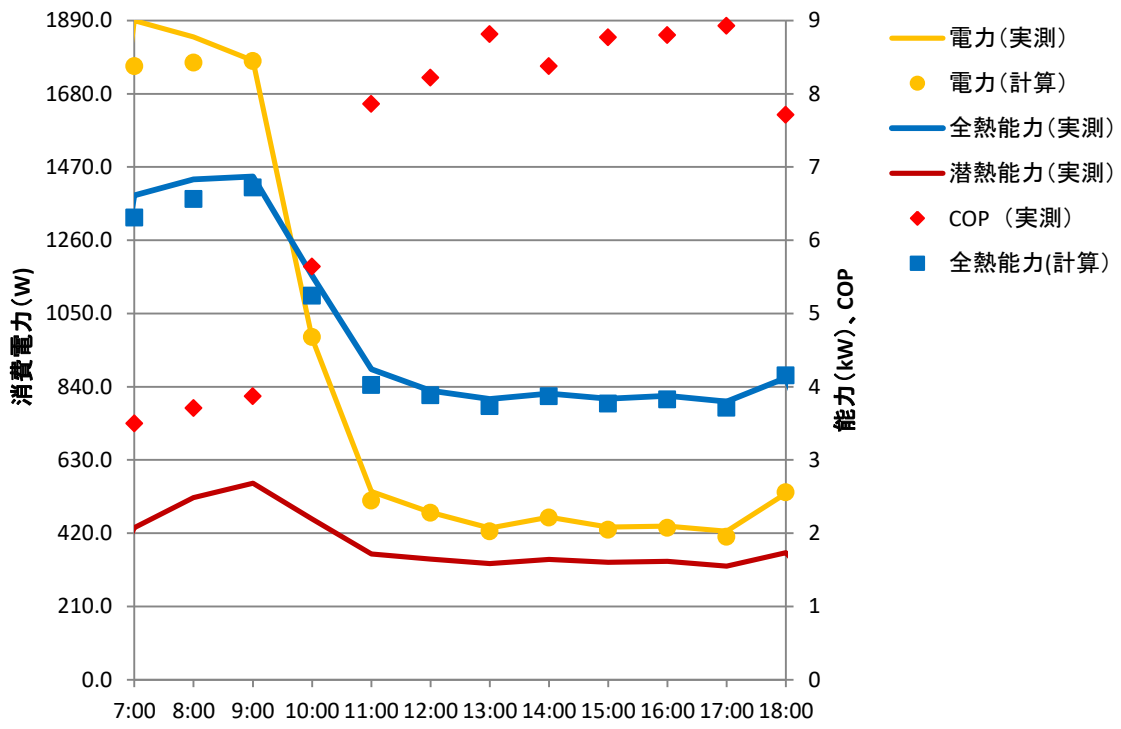


図 159 2/14 運転状態

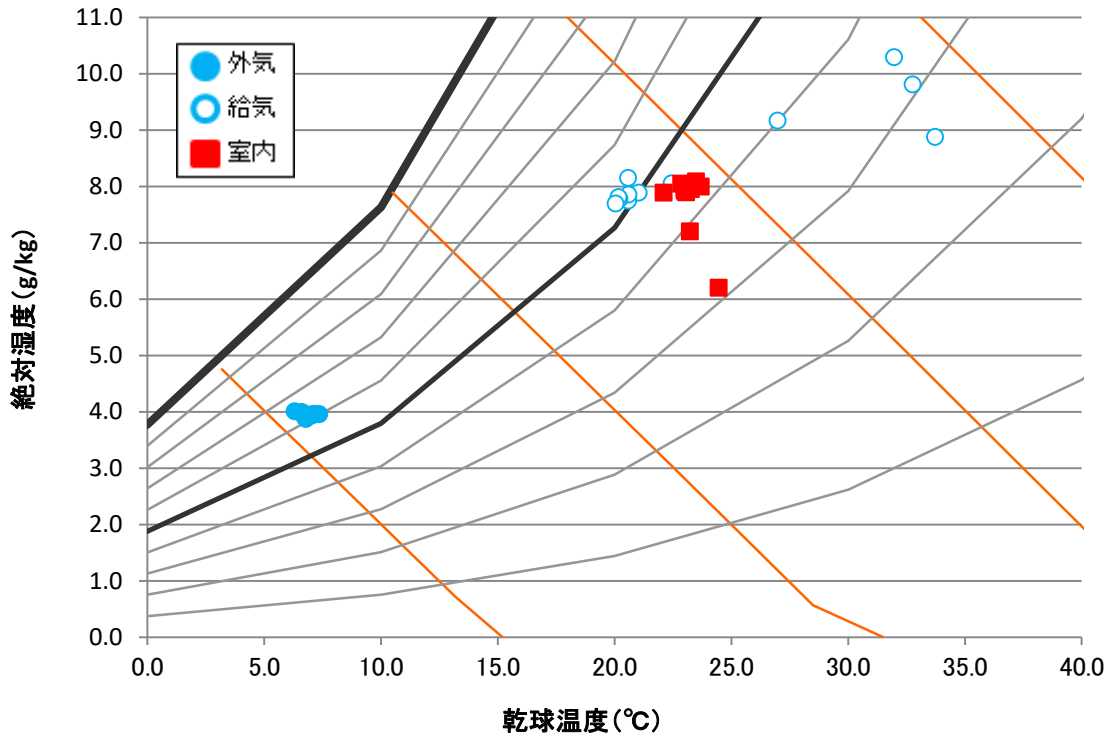


図 160 2/14 空気状態

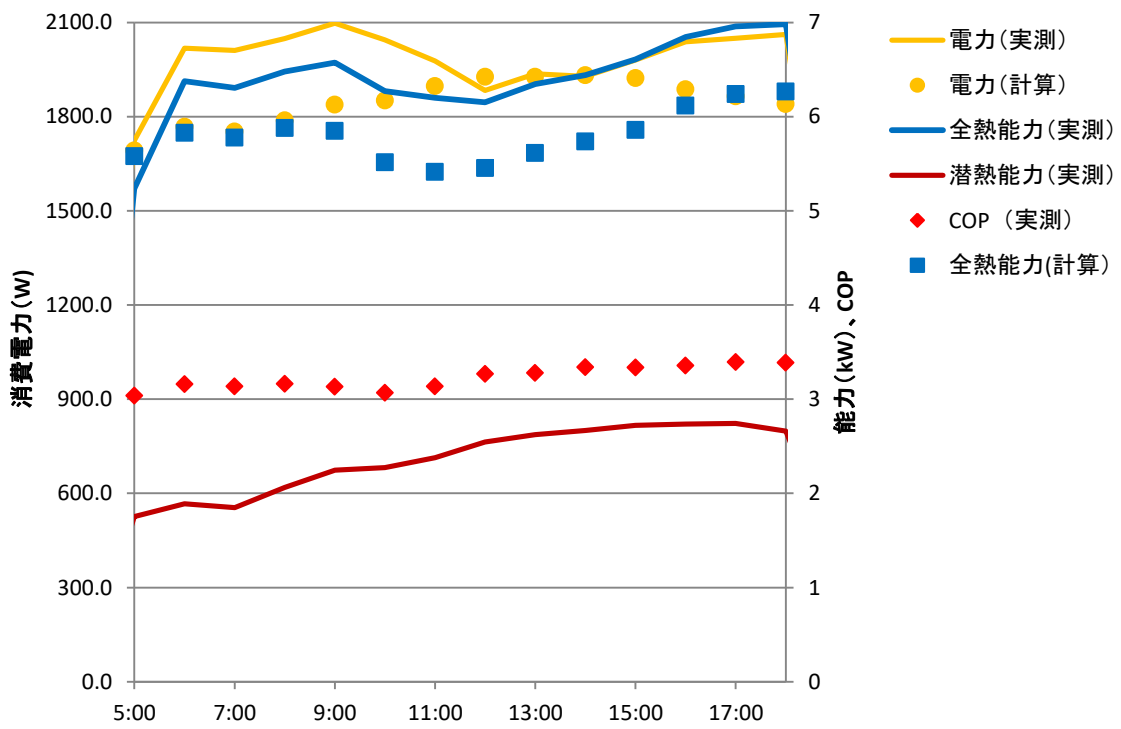


図 161 2/17 運転状態

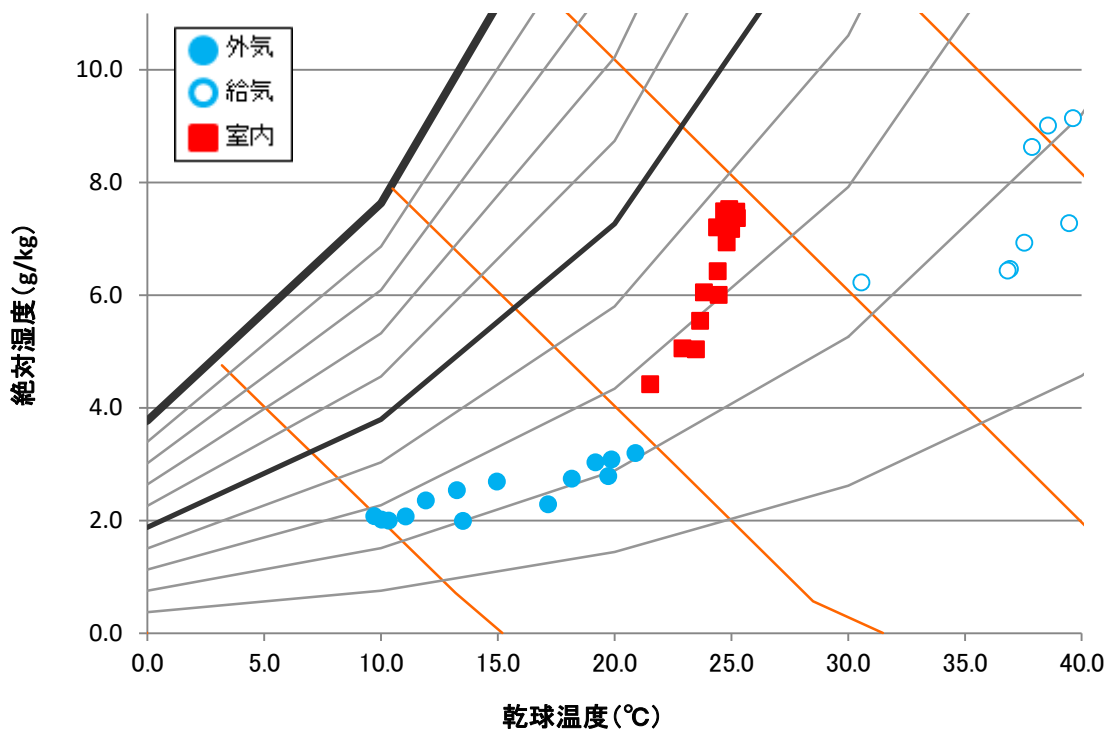


図 162 2/17 空気状態

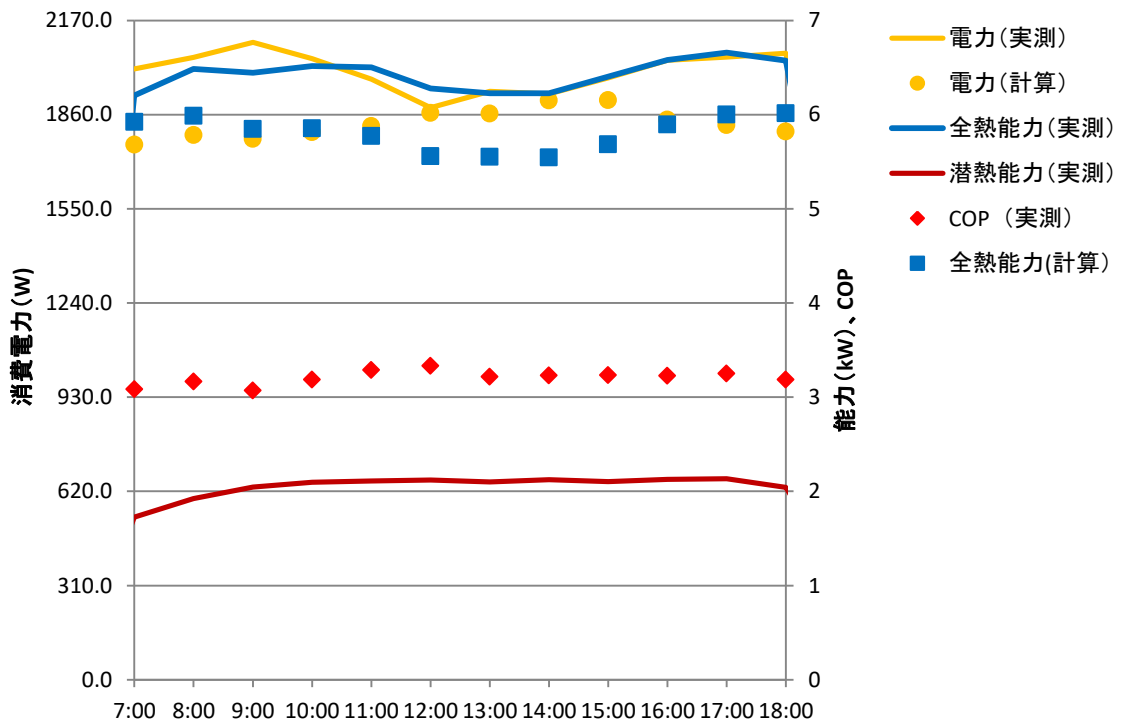


図 163 2/18 運転状態

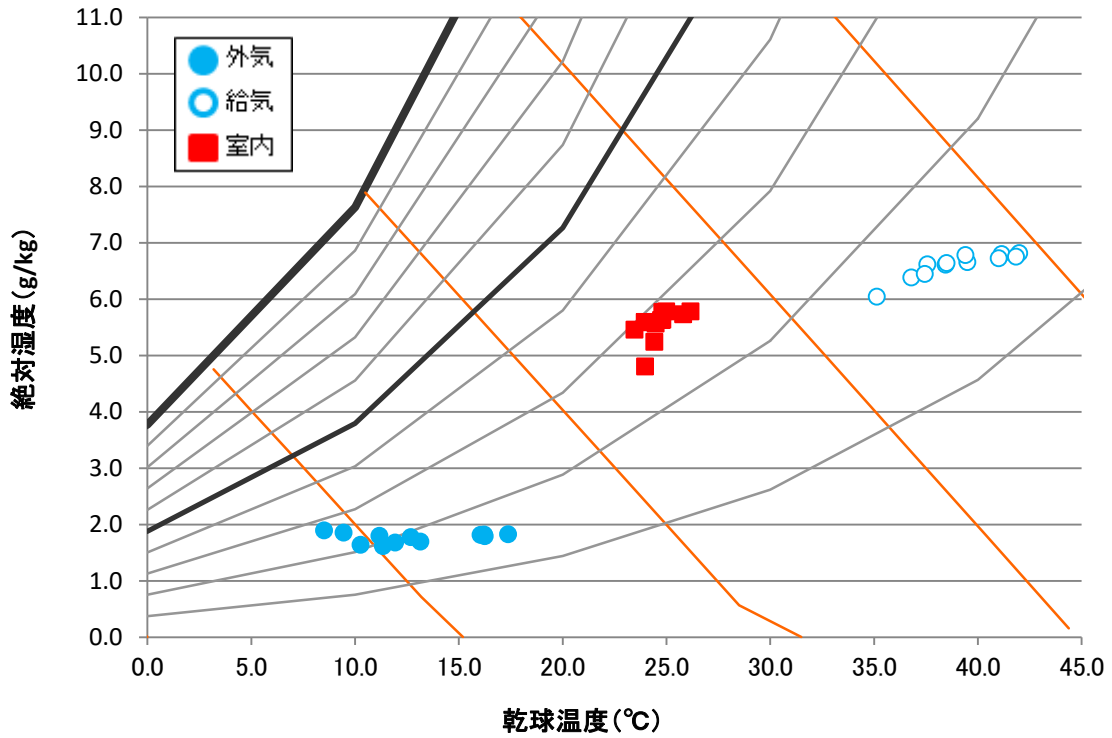


図 164 2/18 空気状態

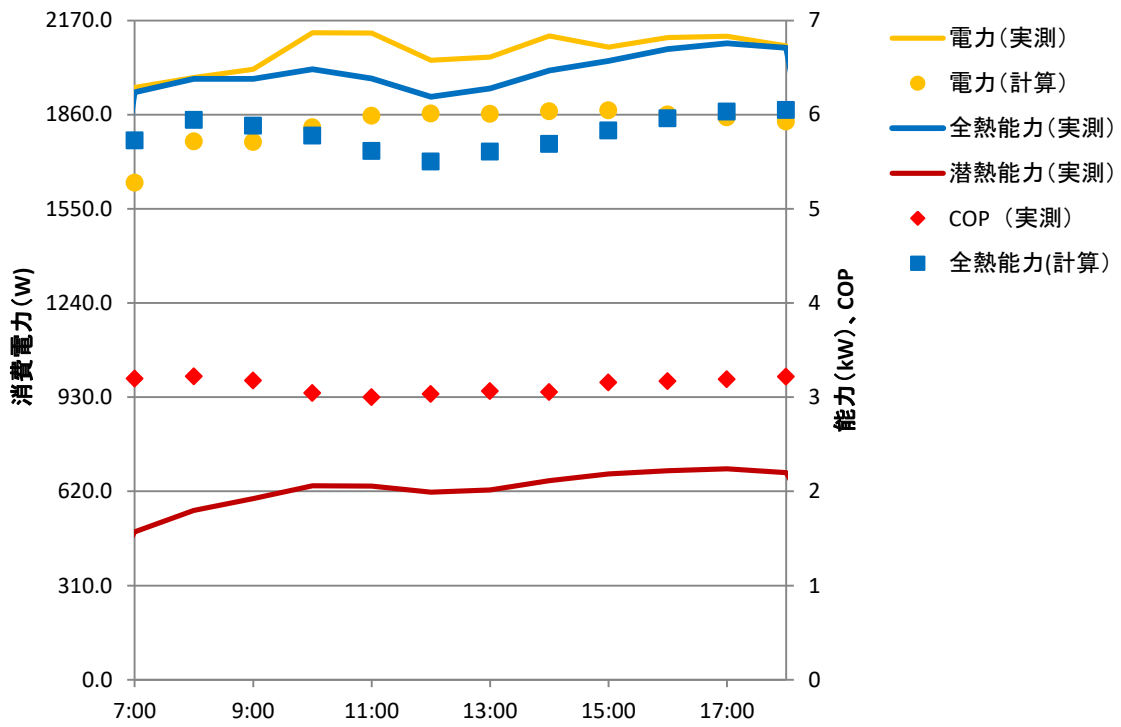


図 165 2/19 運転状態

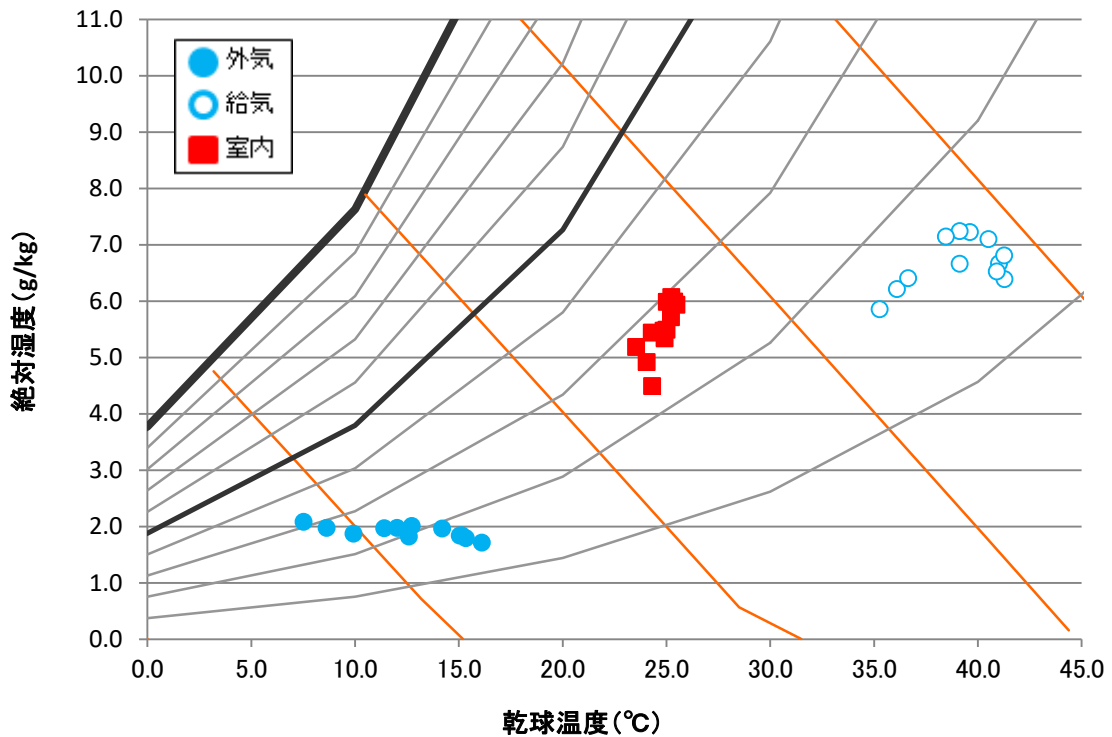


図 166 2/19 空気状態

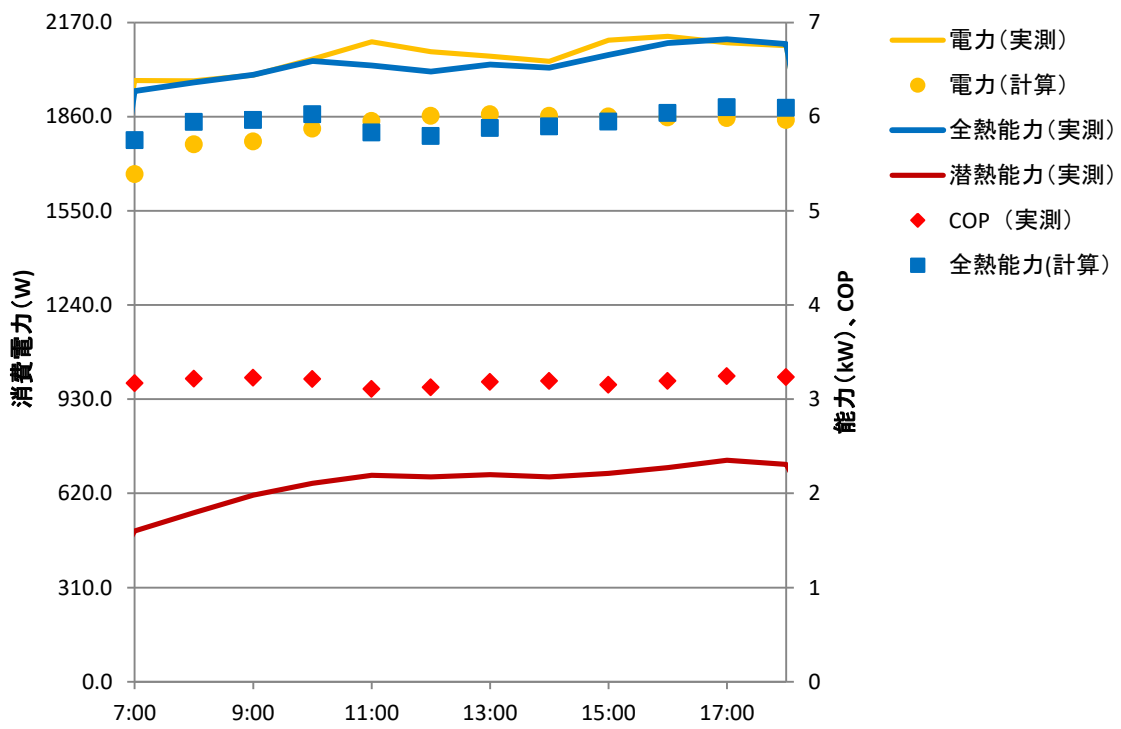


図 167 2/20 運転状態

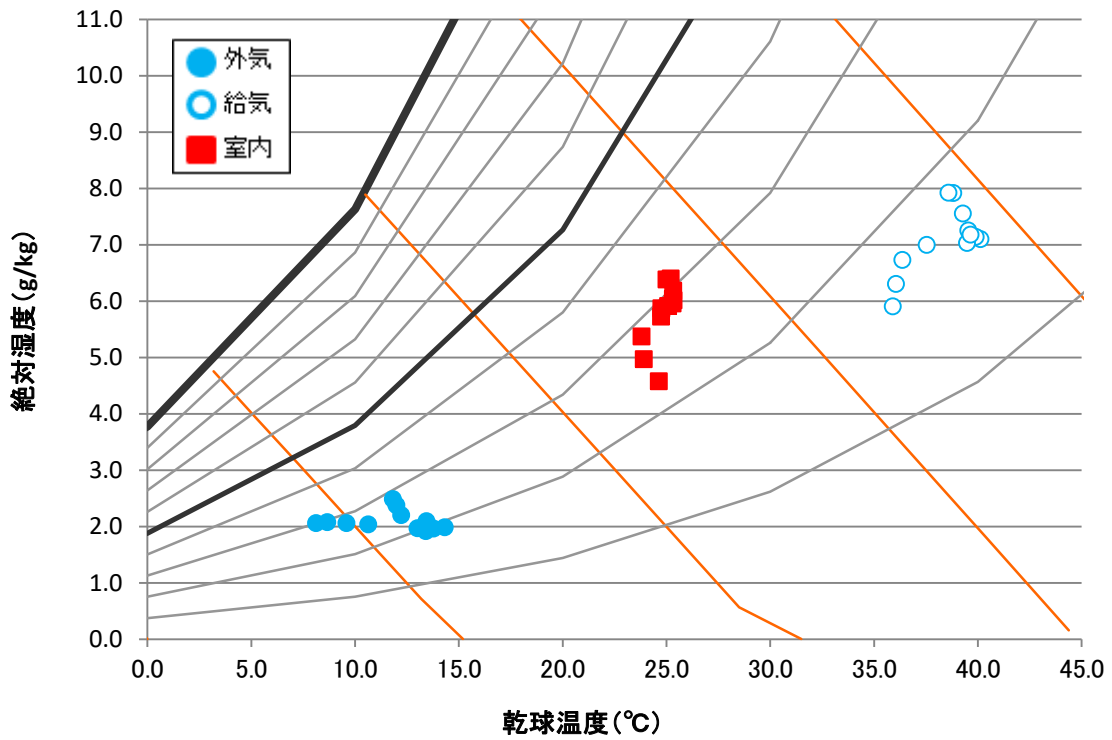


図 168 2/20 空気状態

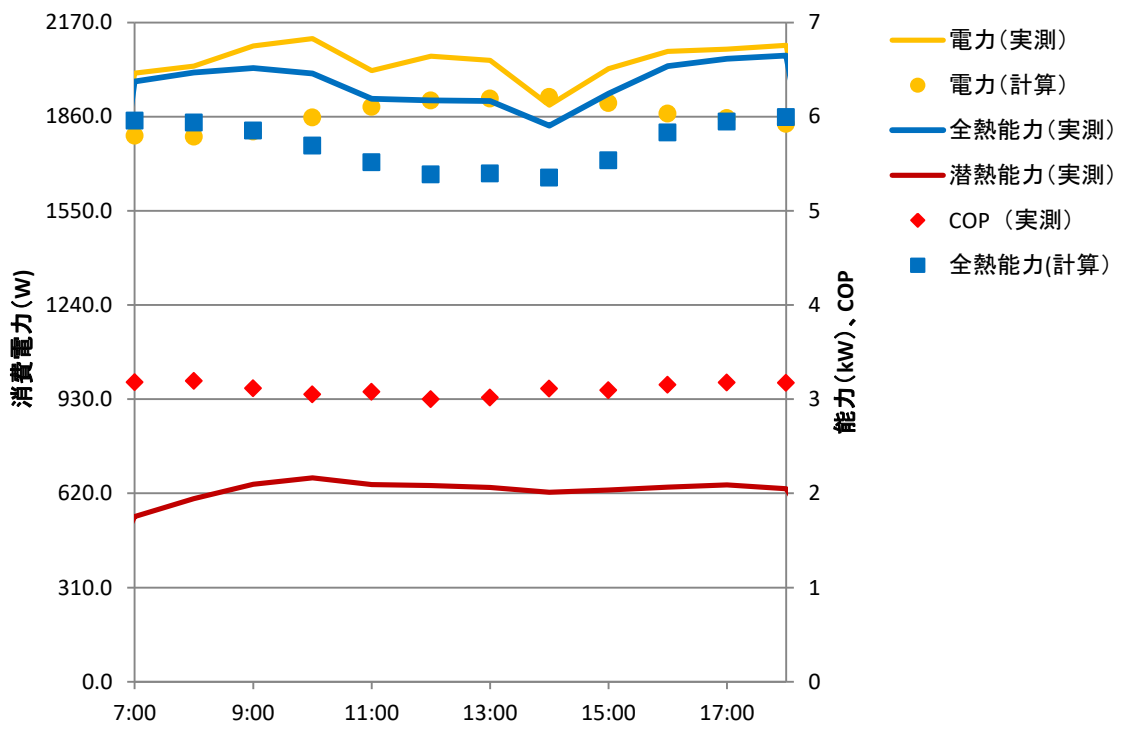


図 169 2/21 運転状況

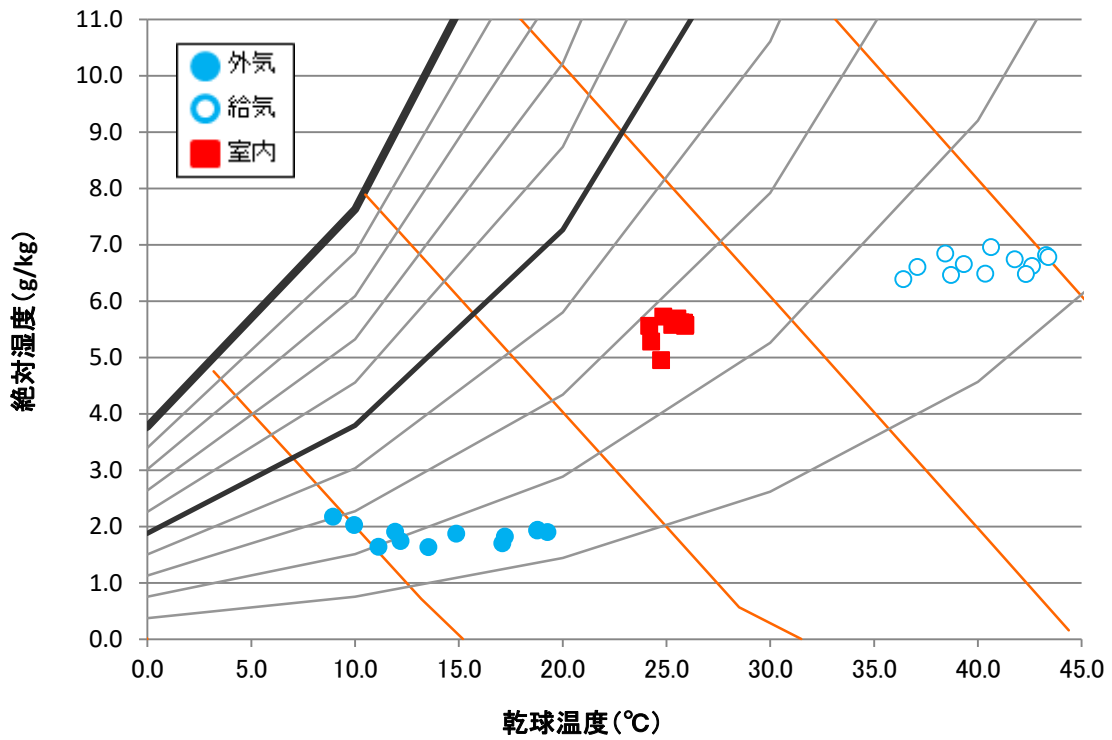


図 170 2/21 空気状況

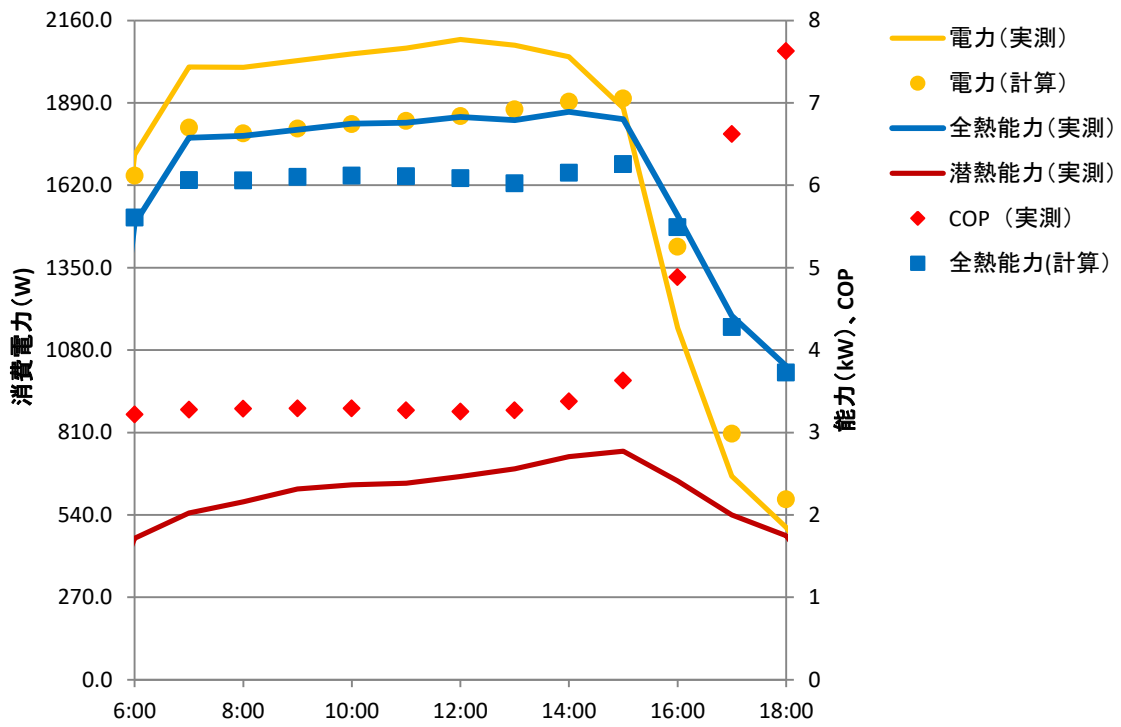


図 171 2/24 運転状況

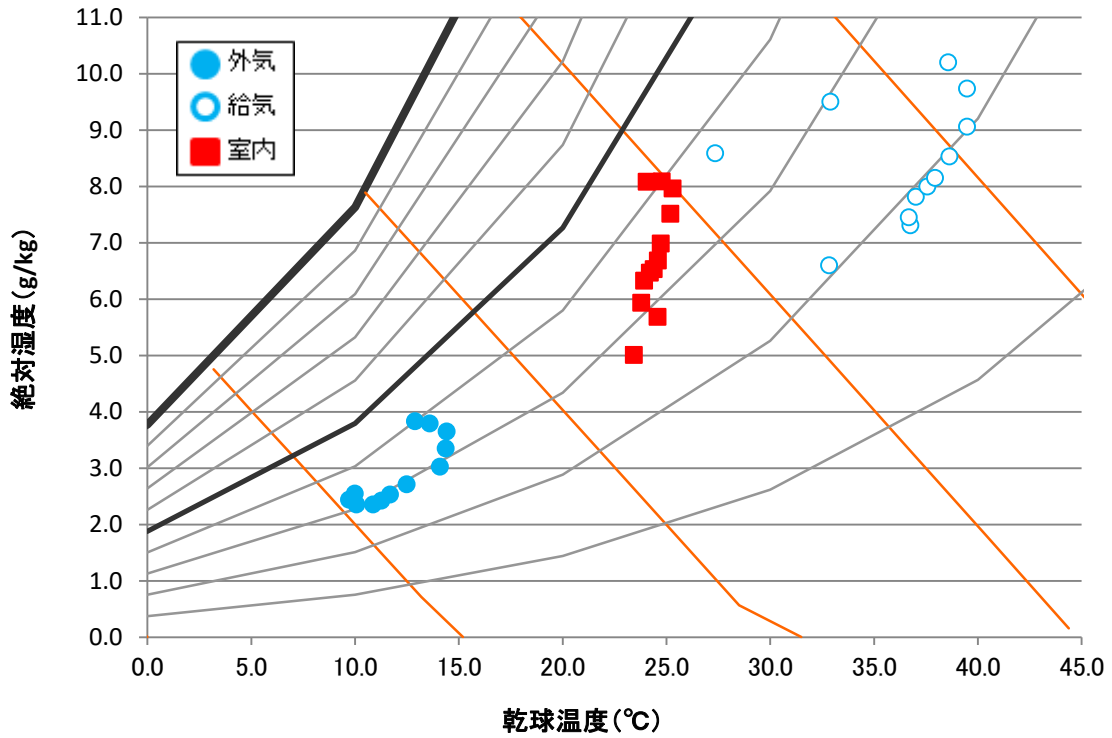


図 172 2/24 空気状況

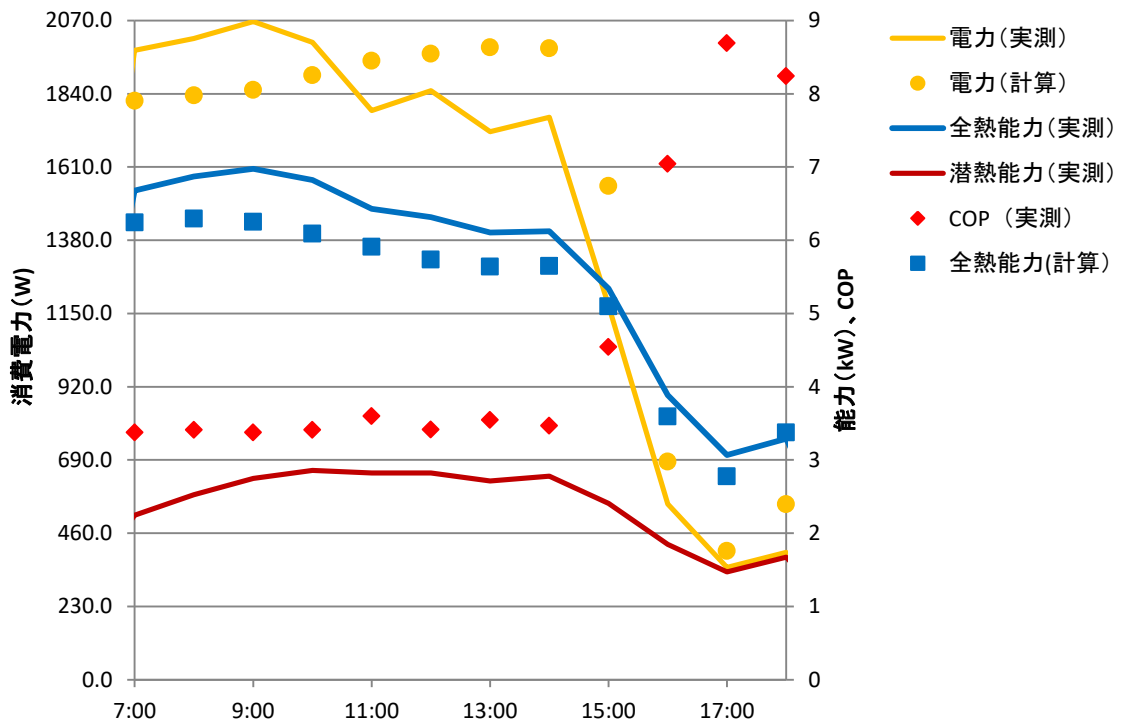


図 173 2/25 運転状況

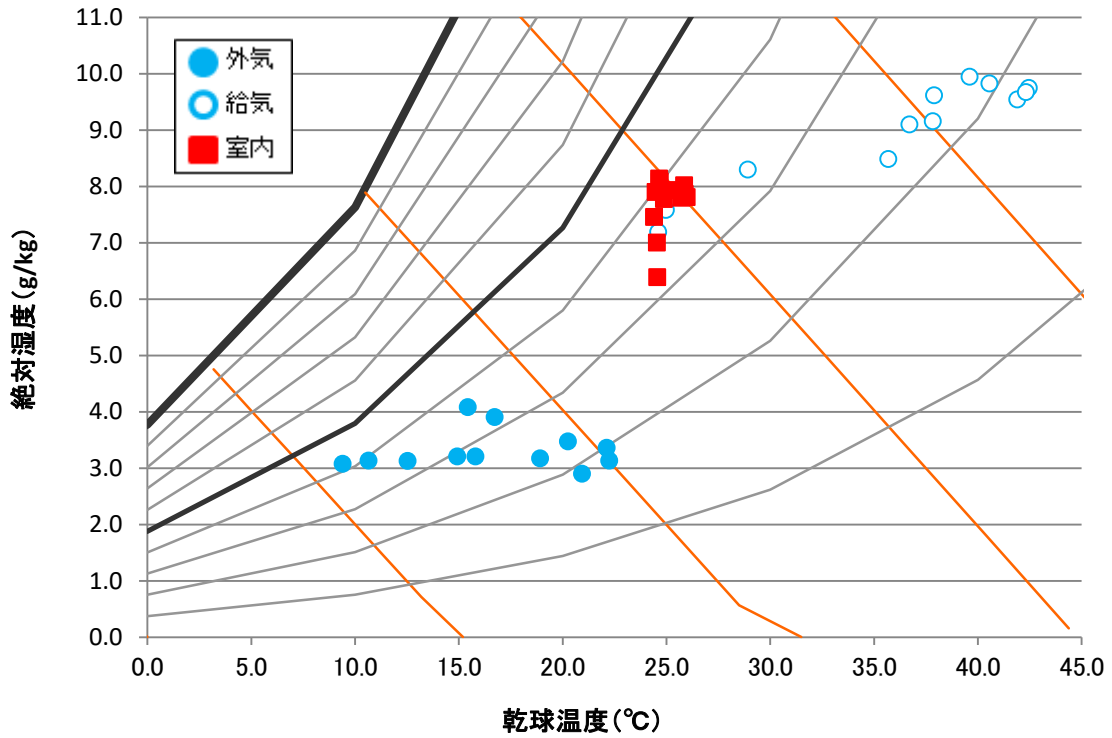


図 174 2/25 空気状況

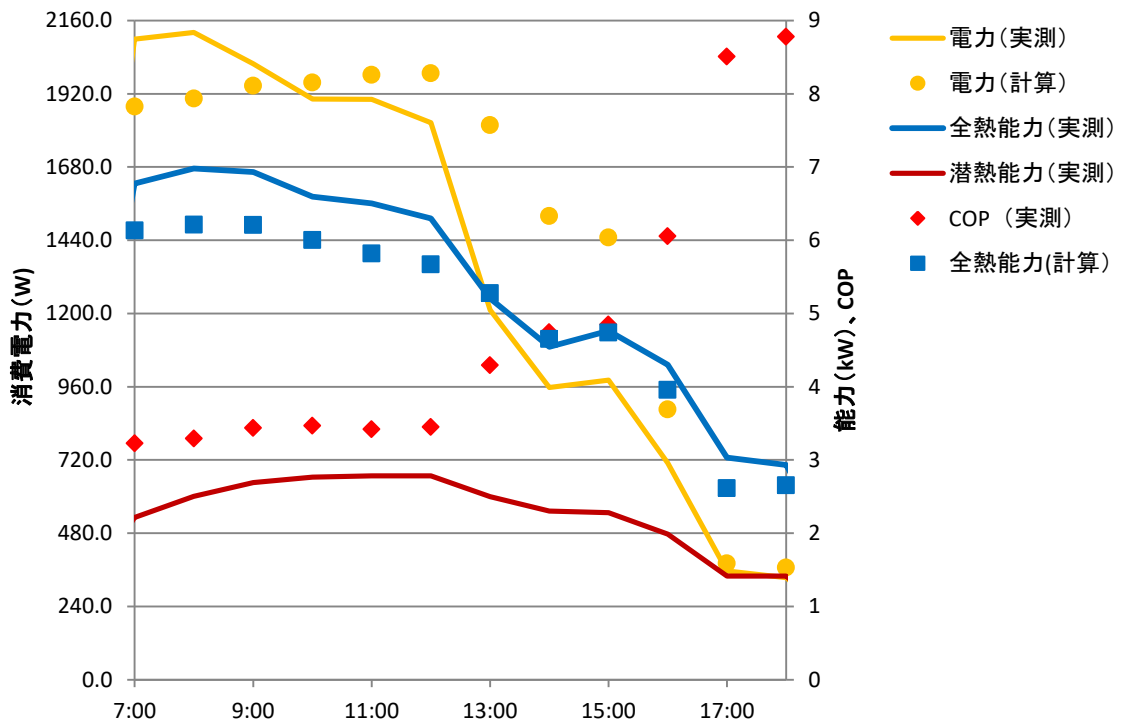


図 175 2/26 運転状況

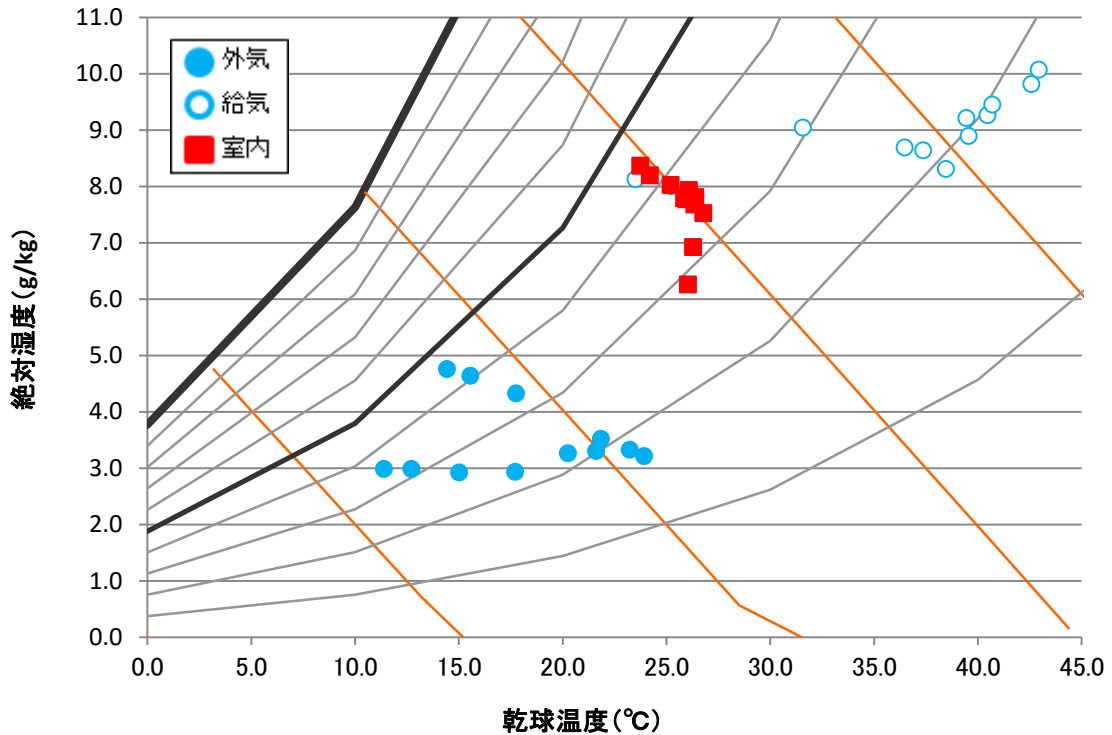


図 176 2/26 空気状況

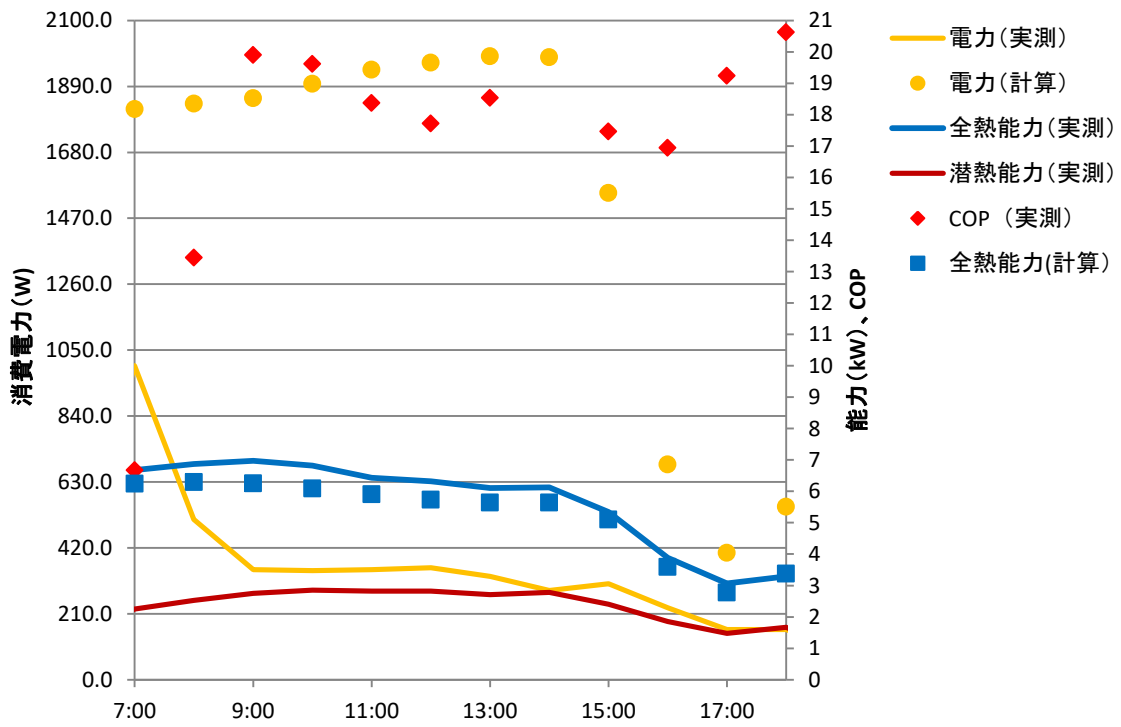


図 177 2/27 運転状況

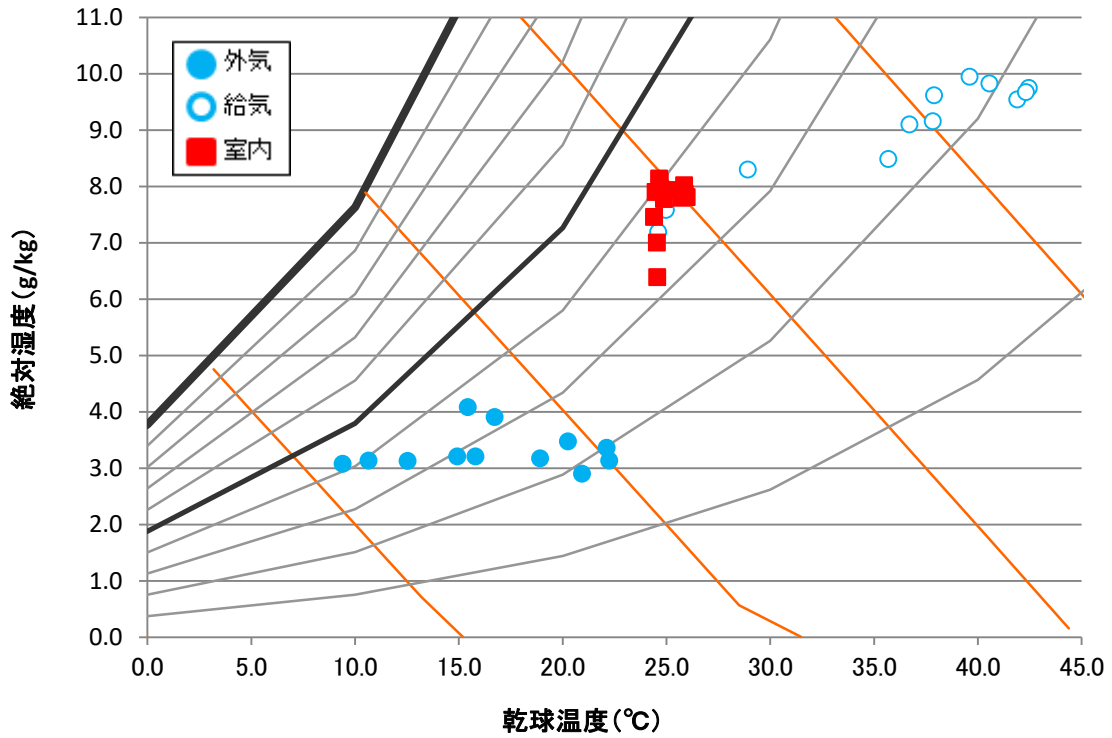


図 178 2/27 空気状況

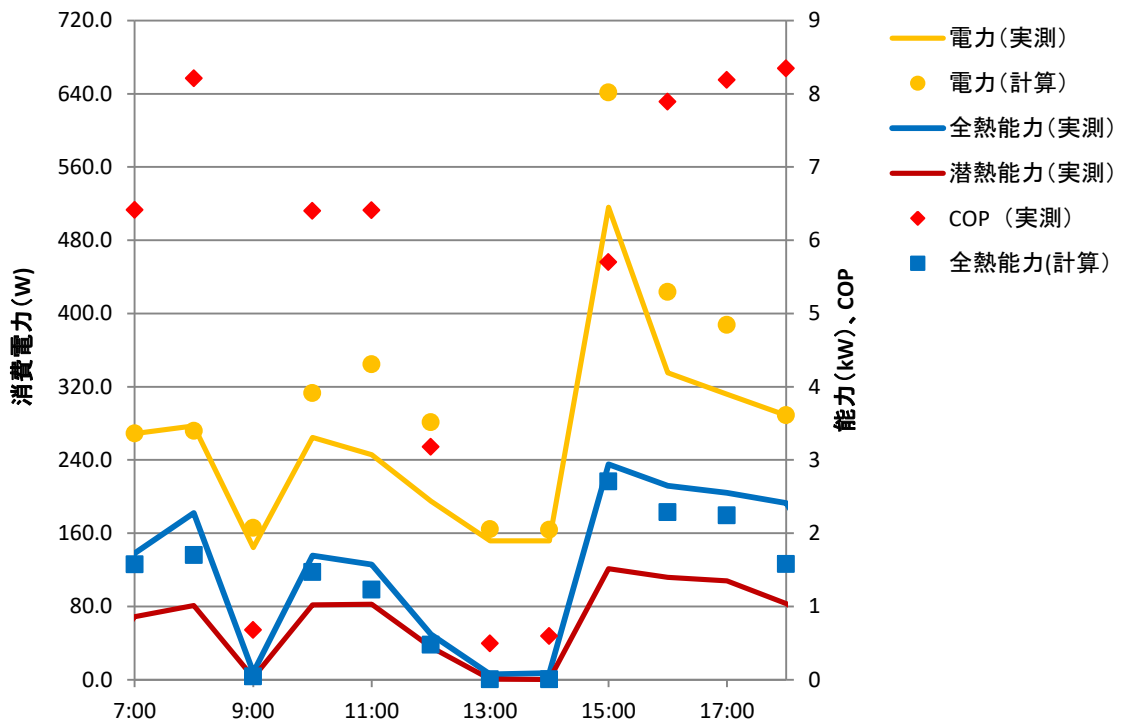


図 179 2/28 運転状況

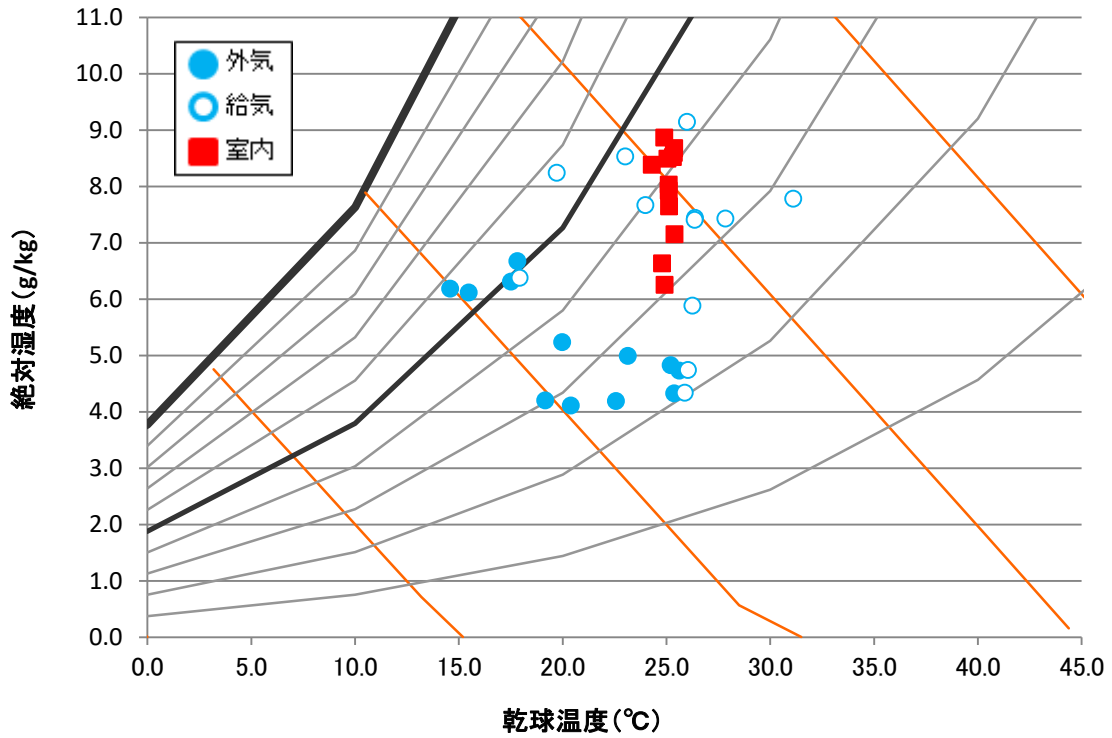


図 180 2/28 空気状況

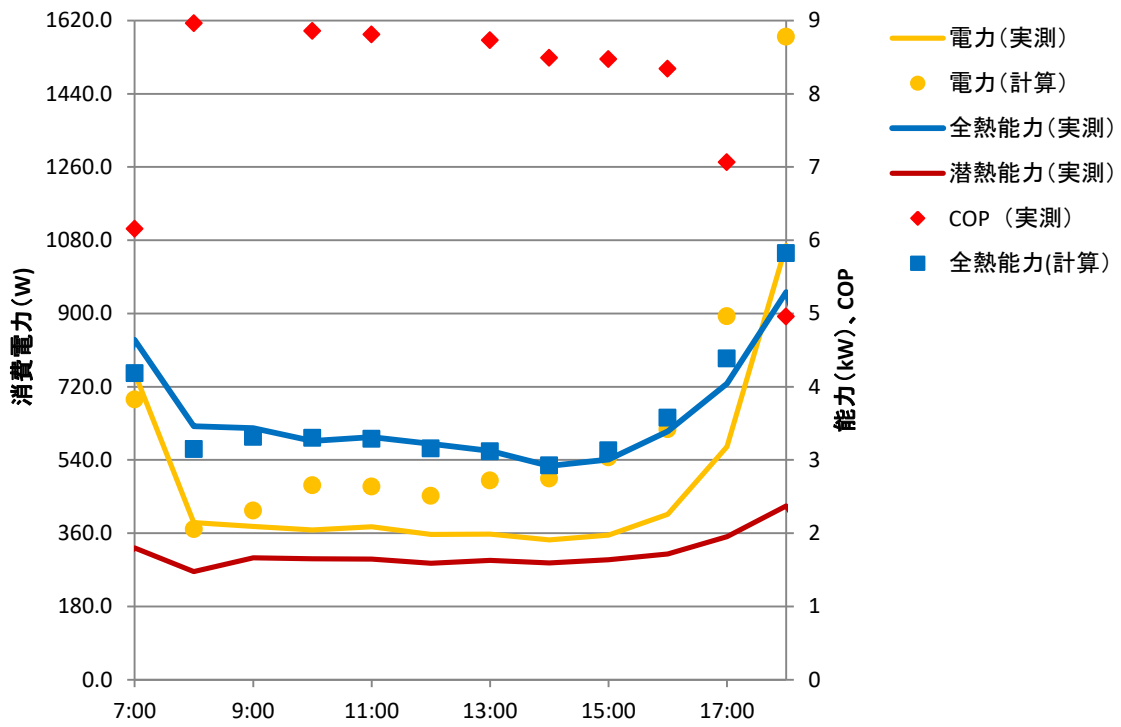


図 181 3/3 運転状況

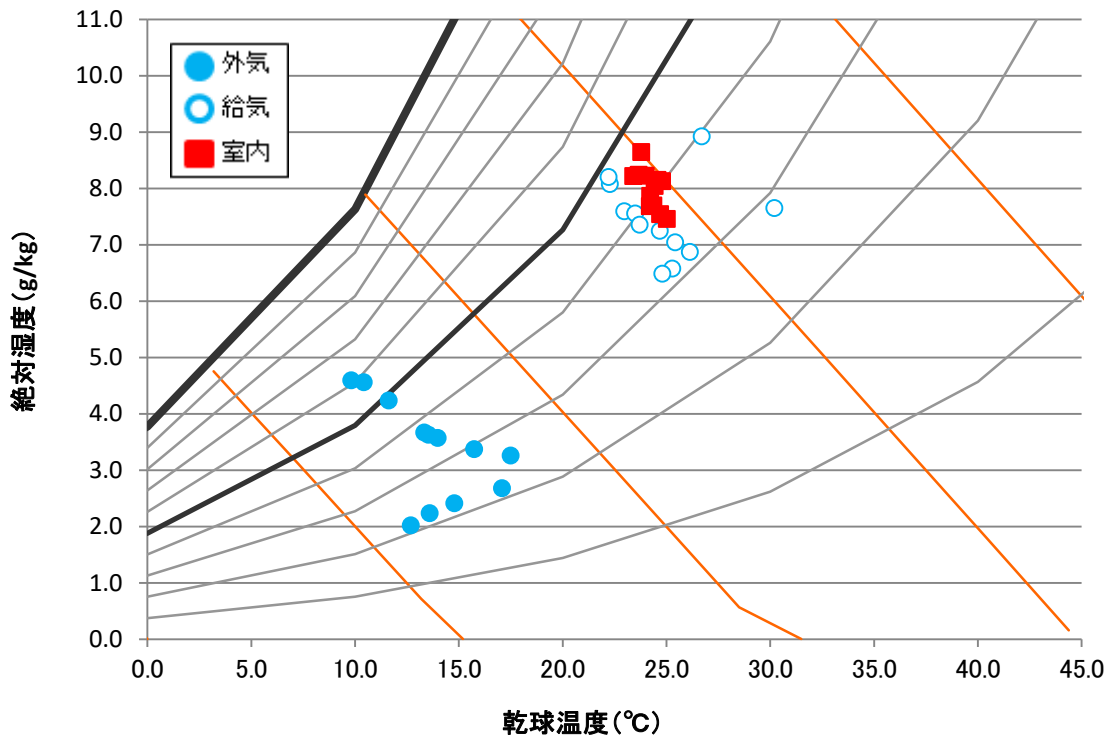


図 182 3/3 空気状況

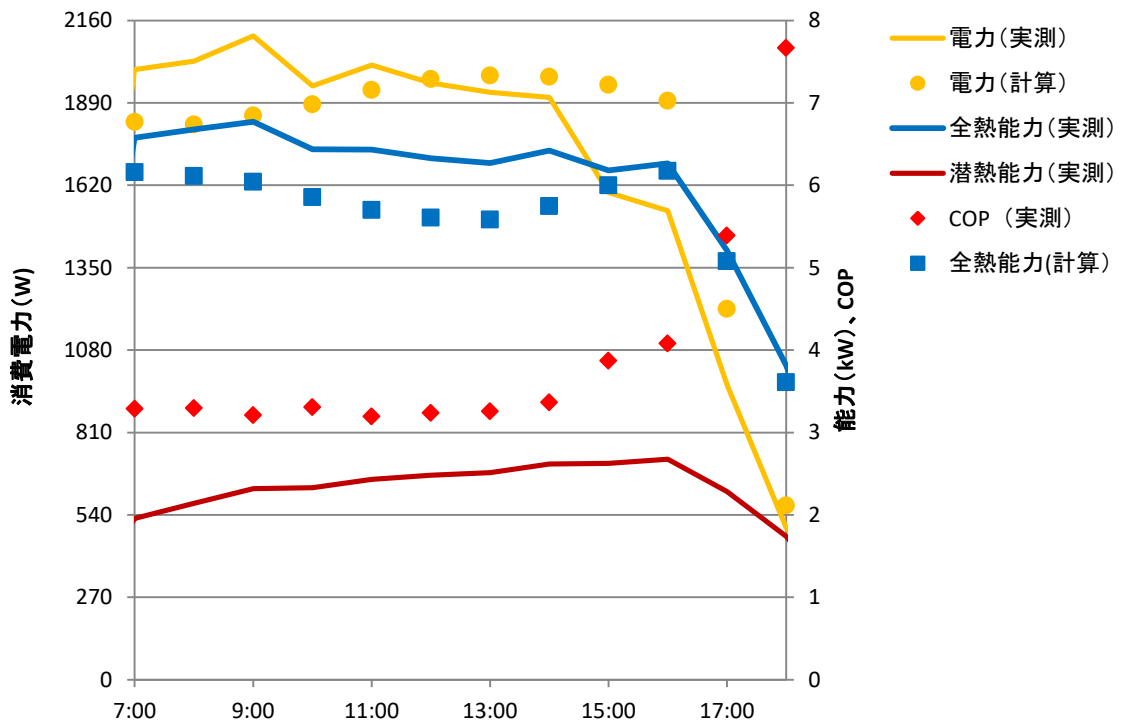


図 183 3/4 運転状況

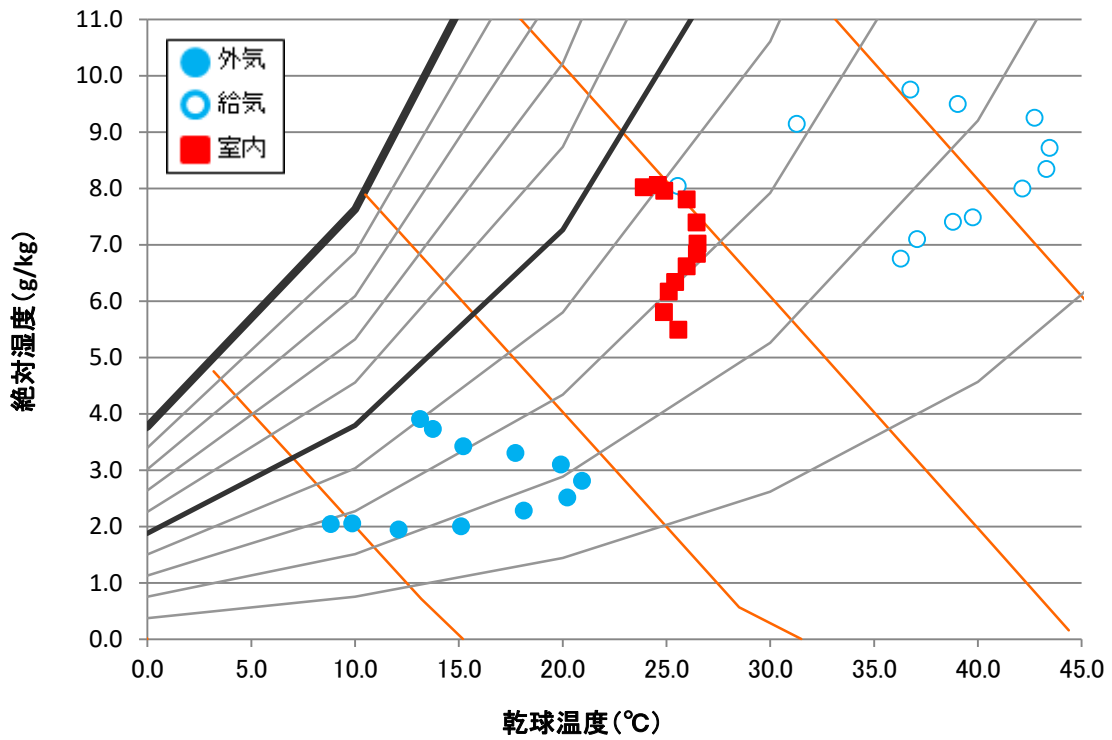


図 184 3/4 空気状況

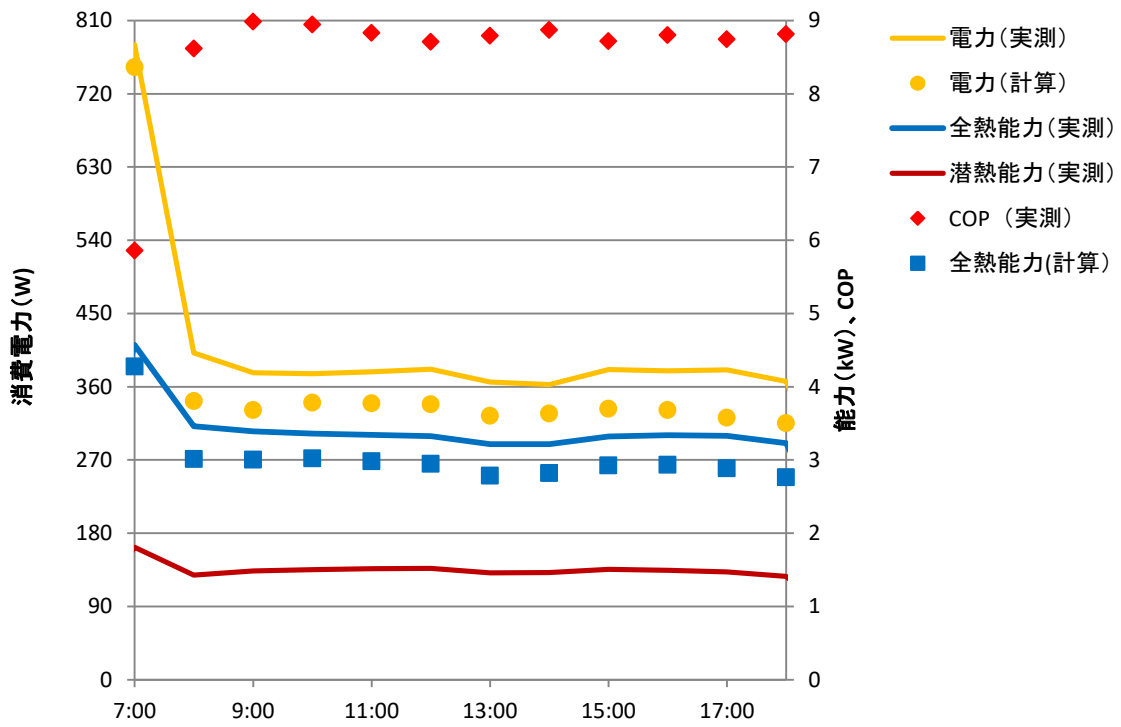


図 185 3/5 運転状況

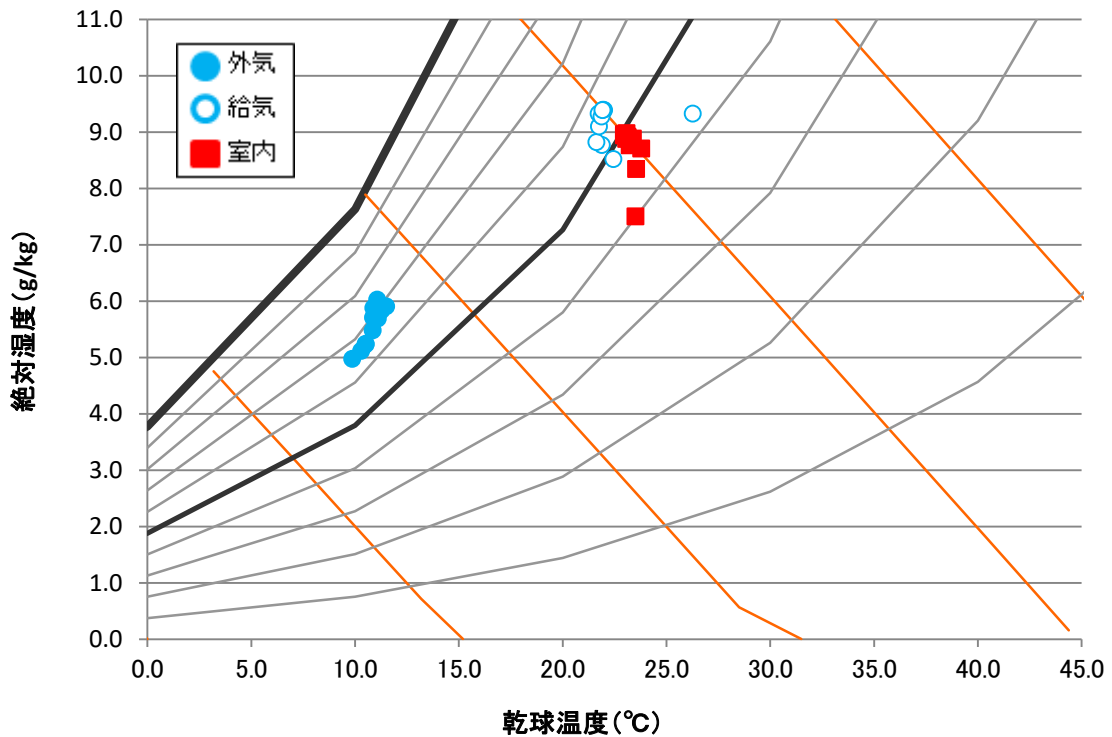


図 186 3/5 空気状況

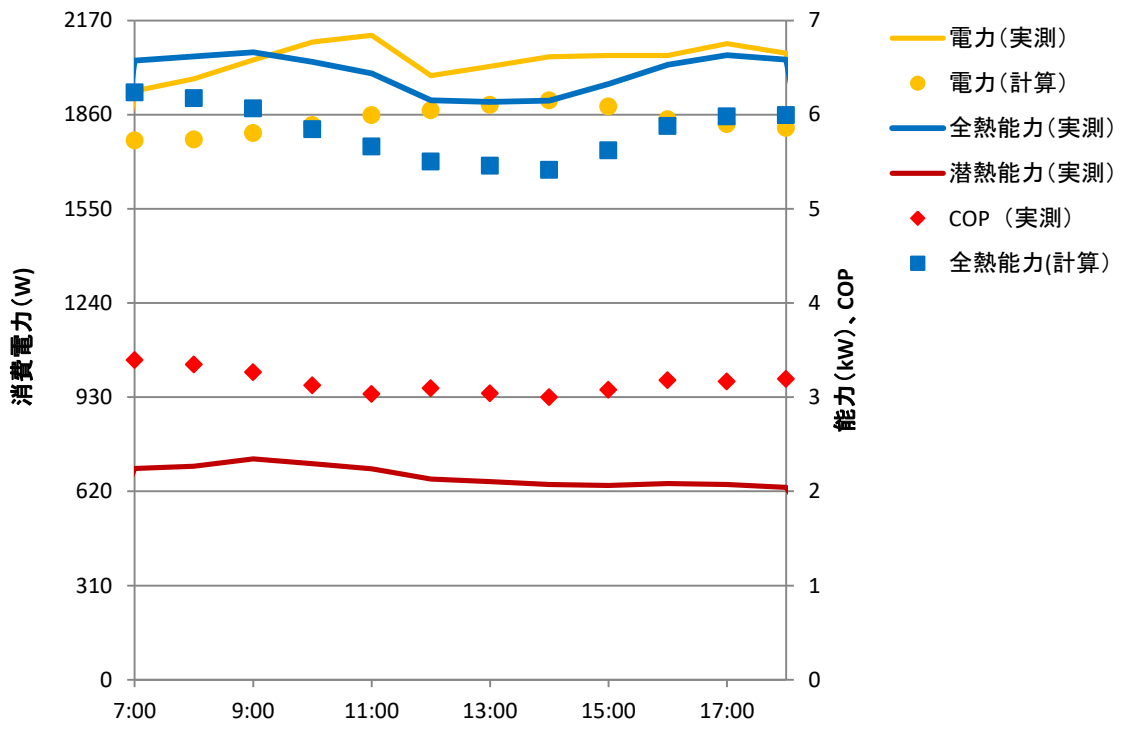


図 187 3/6 運転状況

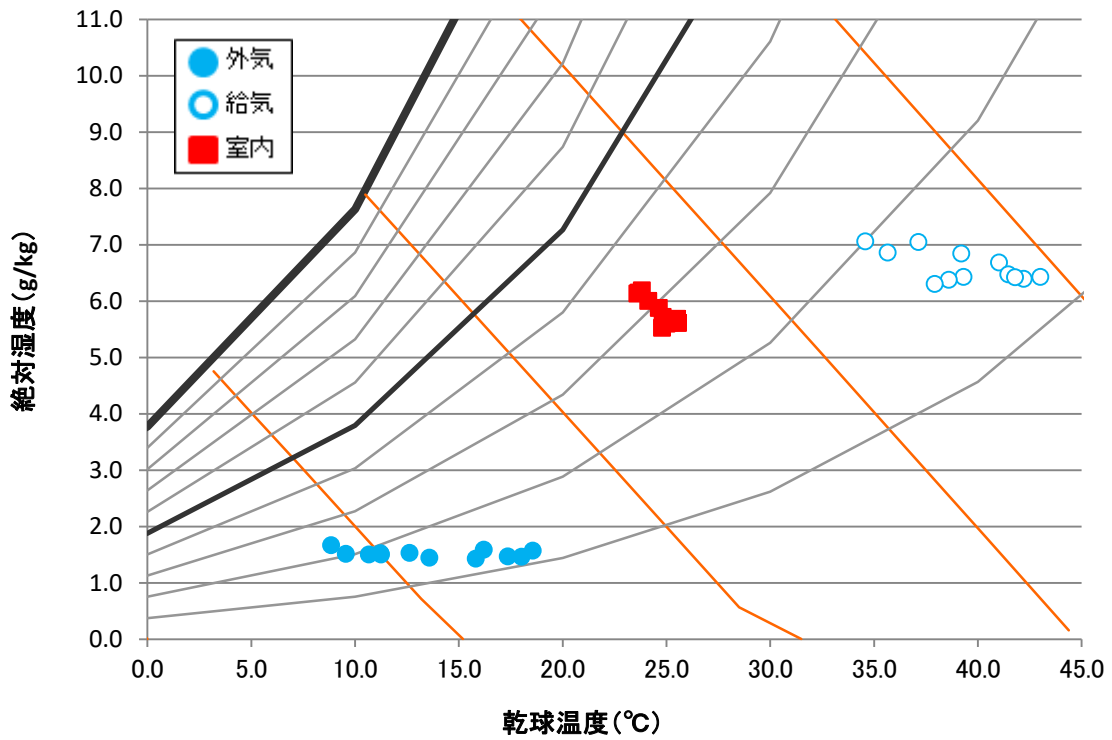


図 188 3/6 空気状況

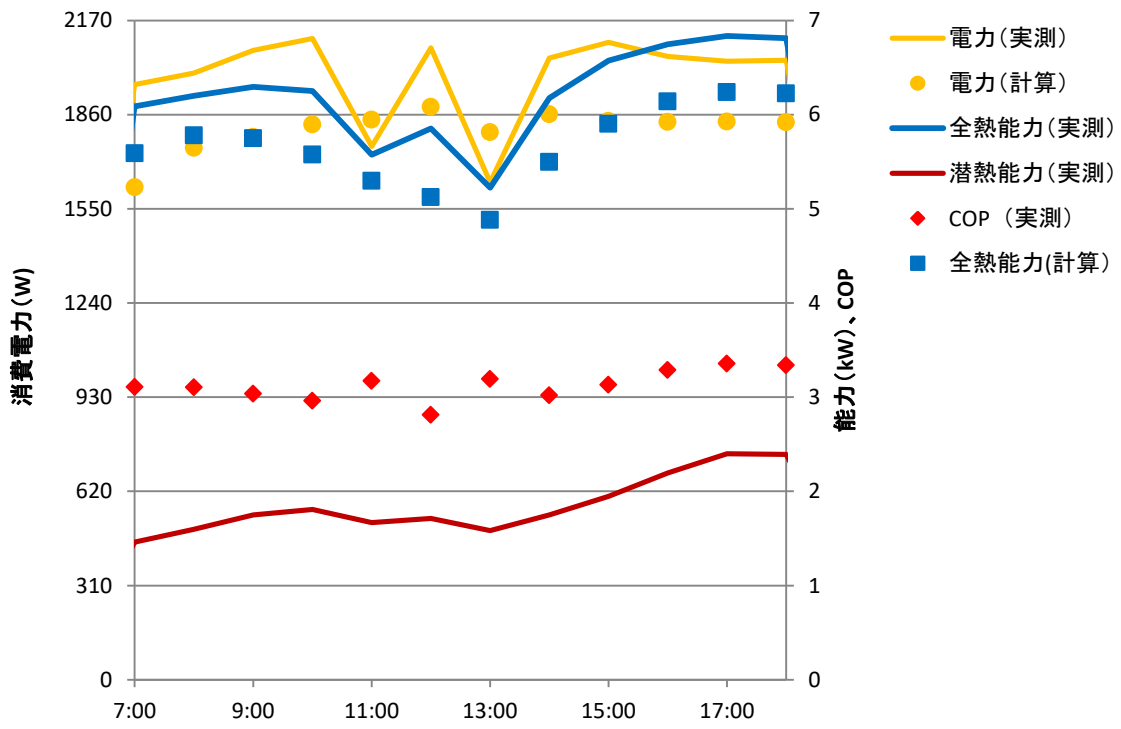


図 189 3/7 運転状況

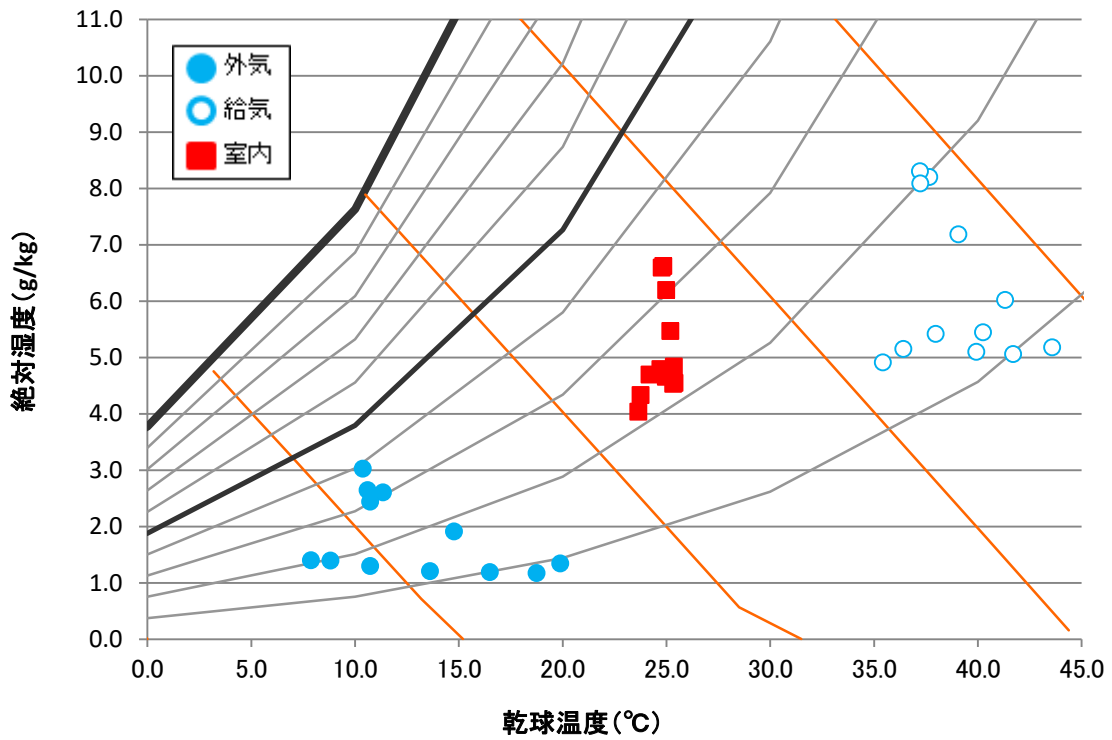


図 190 3/7 空気状況

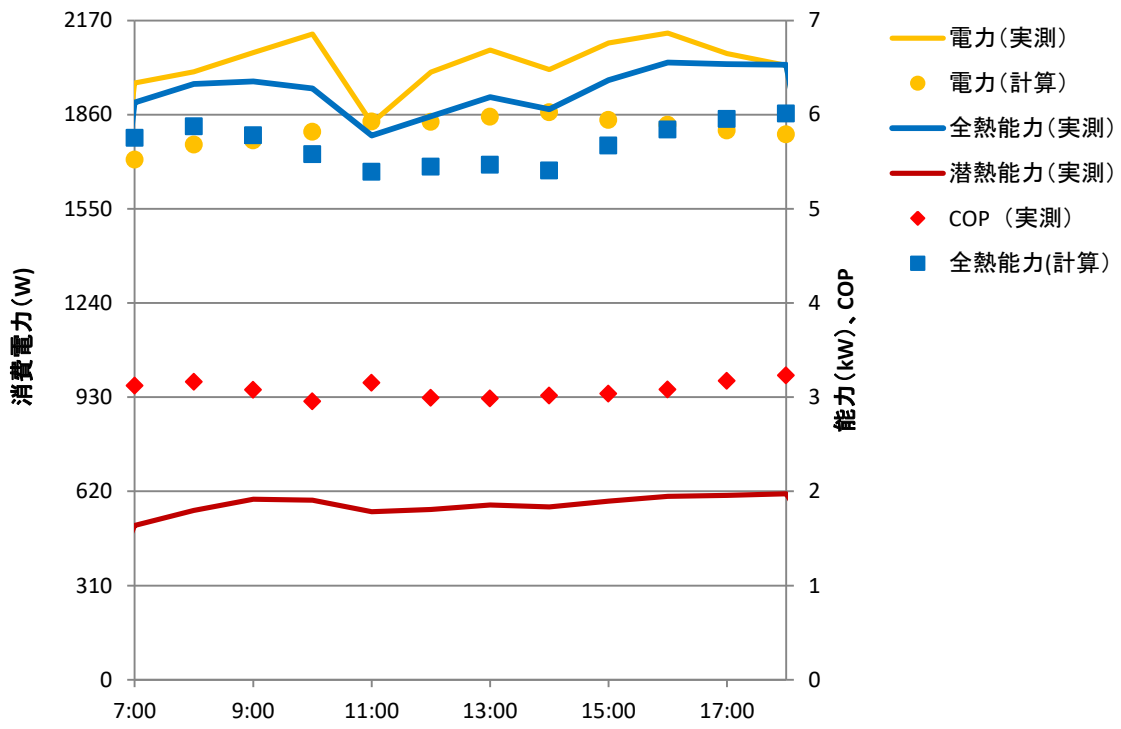


図 191 3/10 運転状況

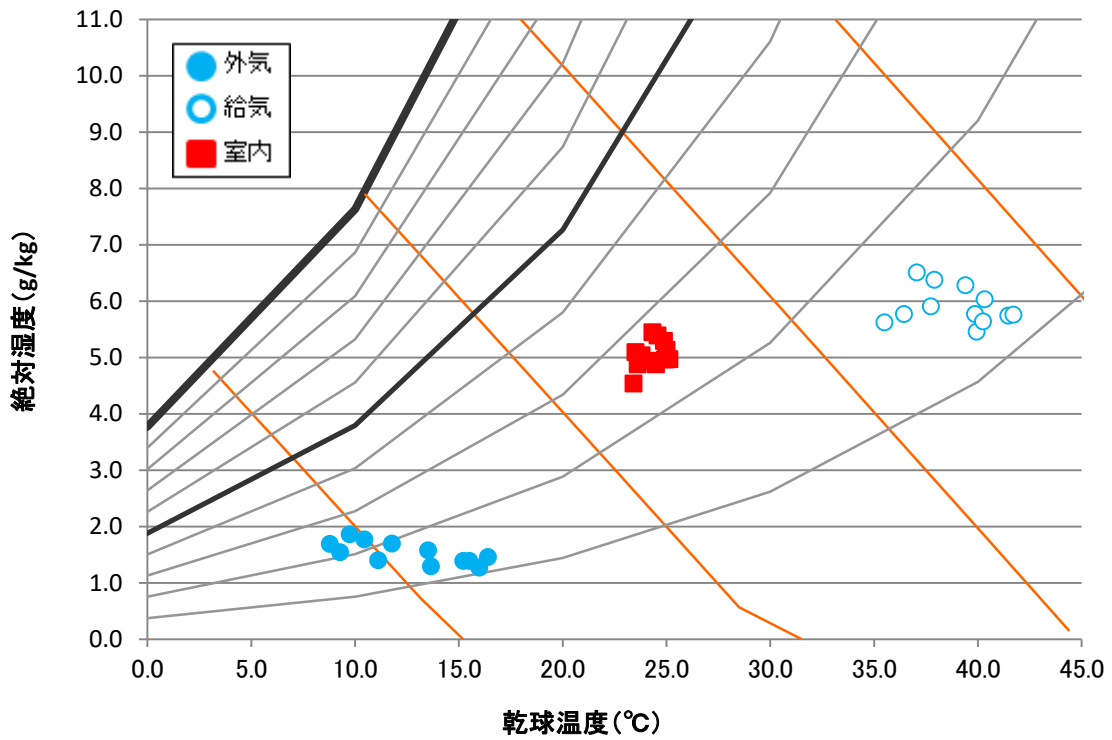


図 192 3/10 空気条件

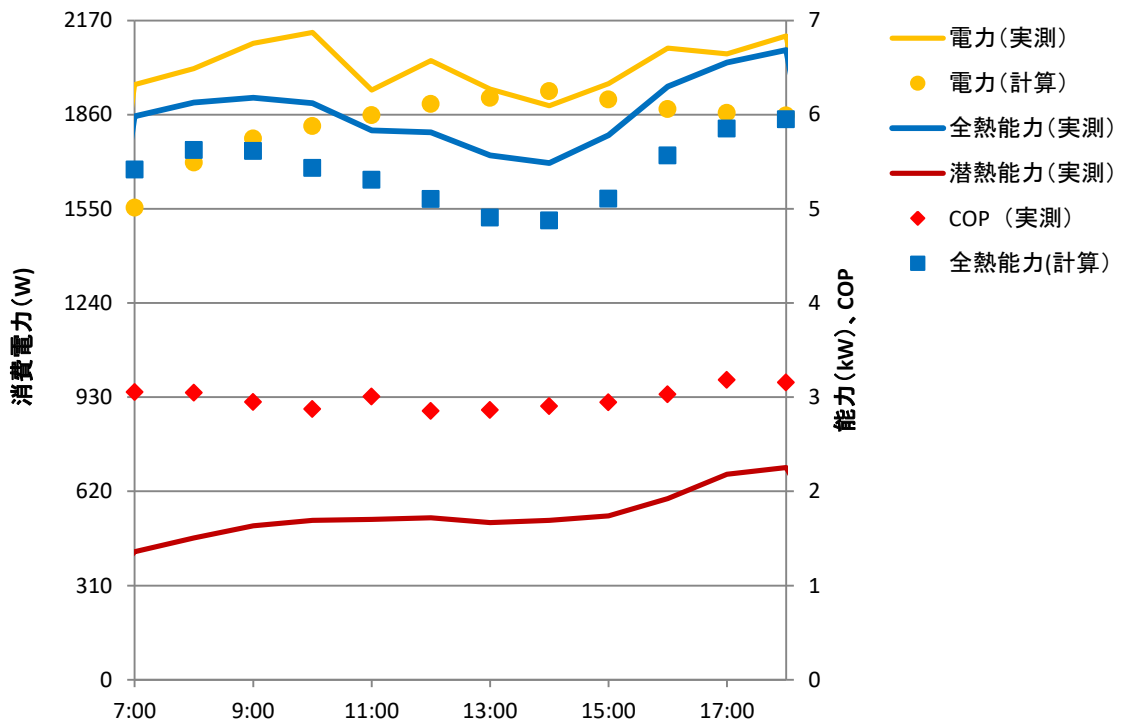


図 193 3/11 運転状況

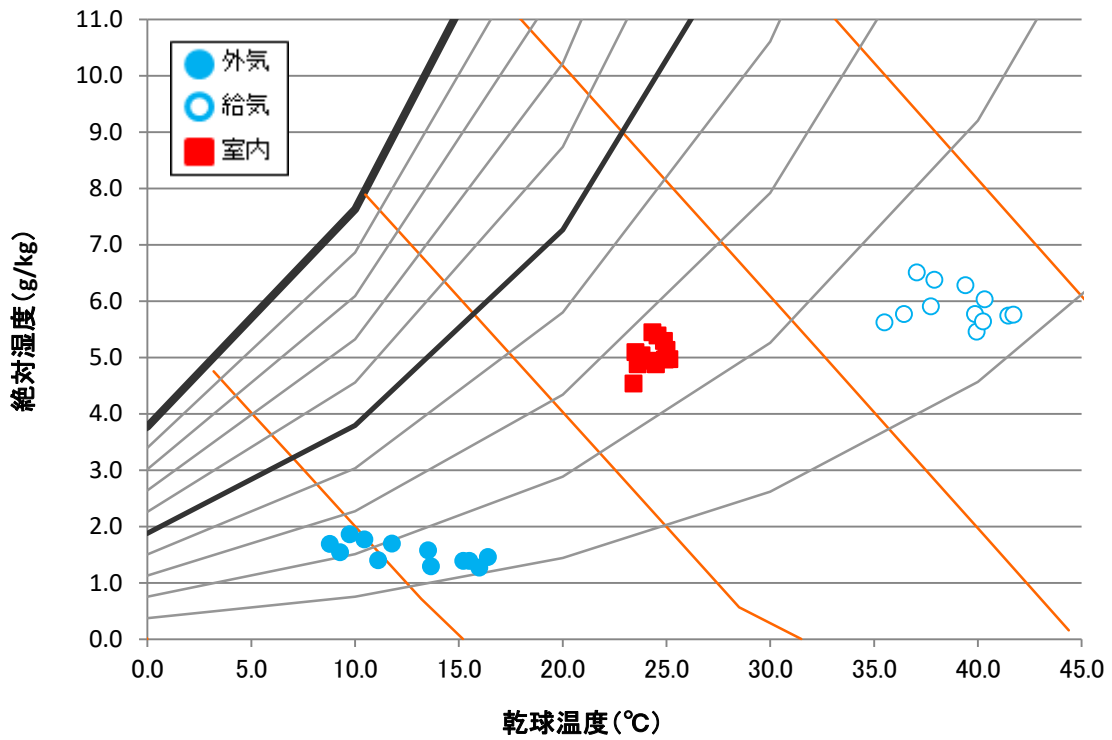


図 194 3/11 空気状況

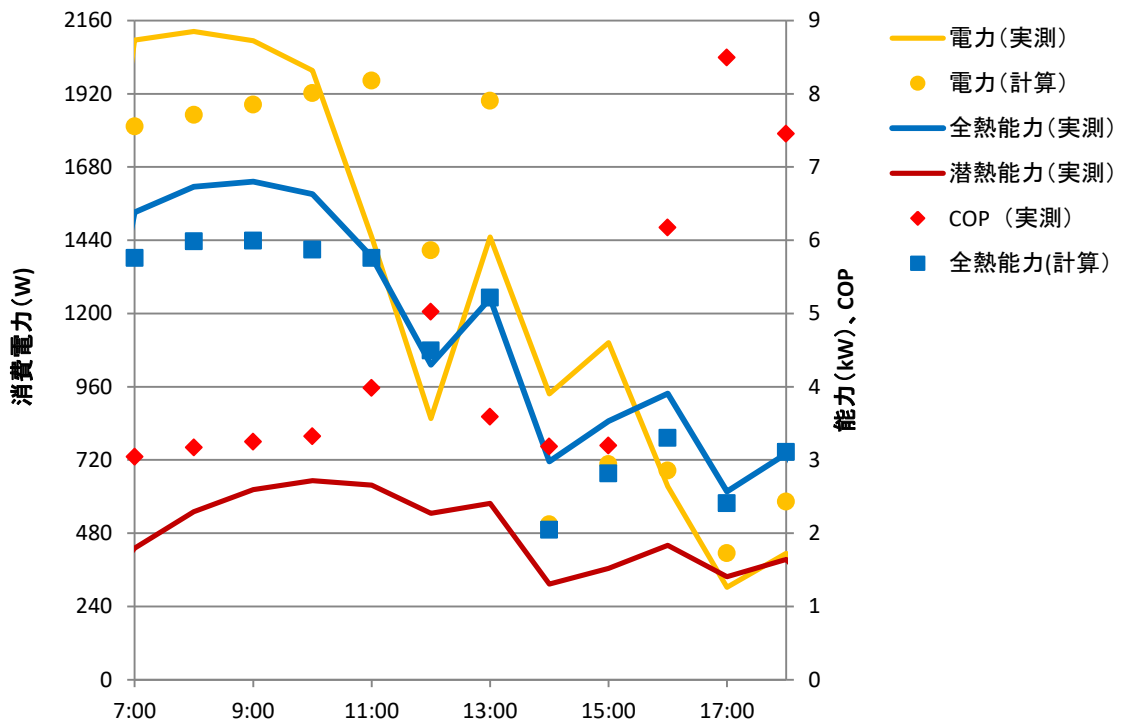


図 195 3/12 運転状況

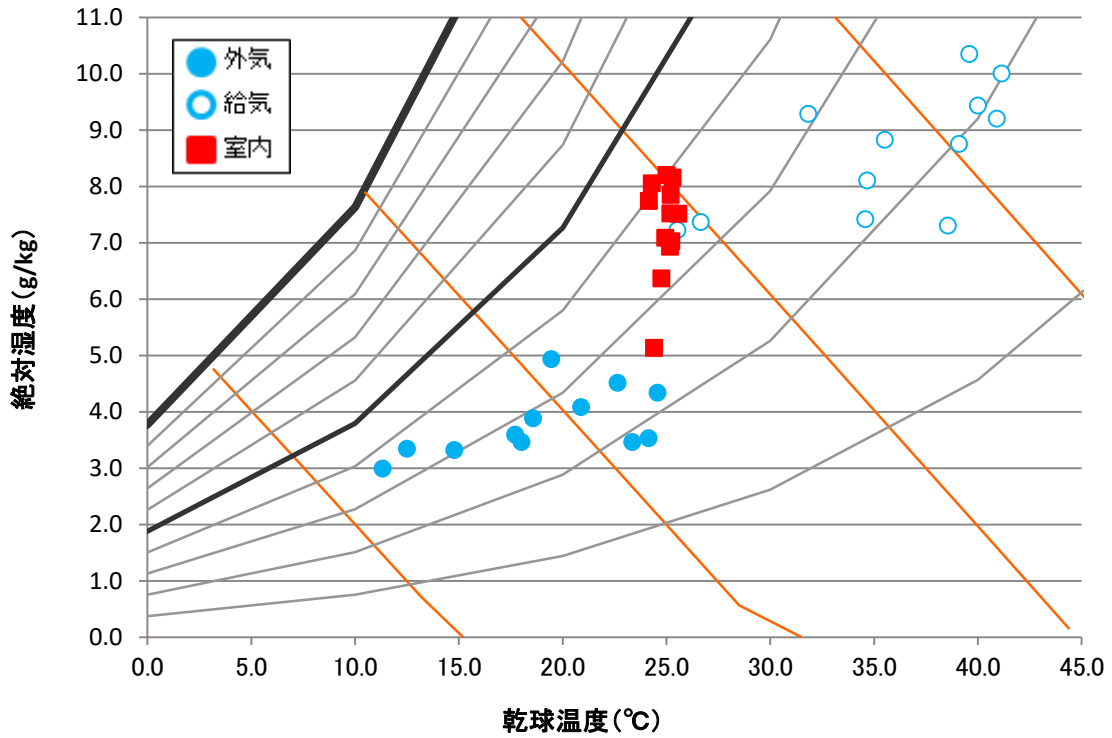


図 196 3/12 空気状況

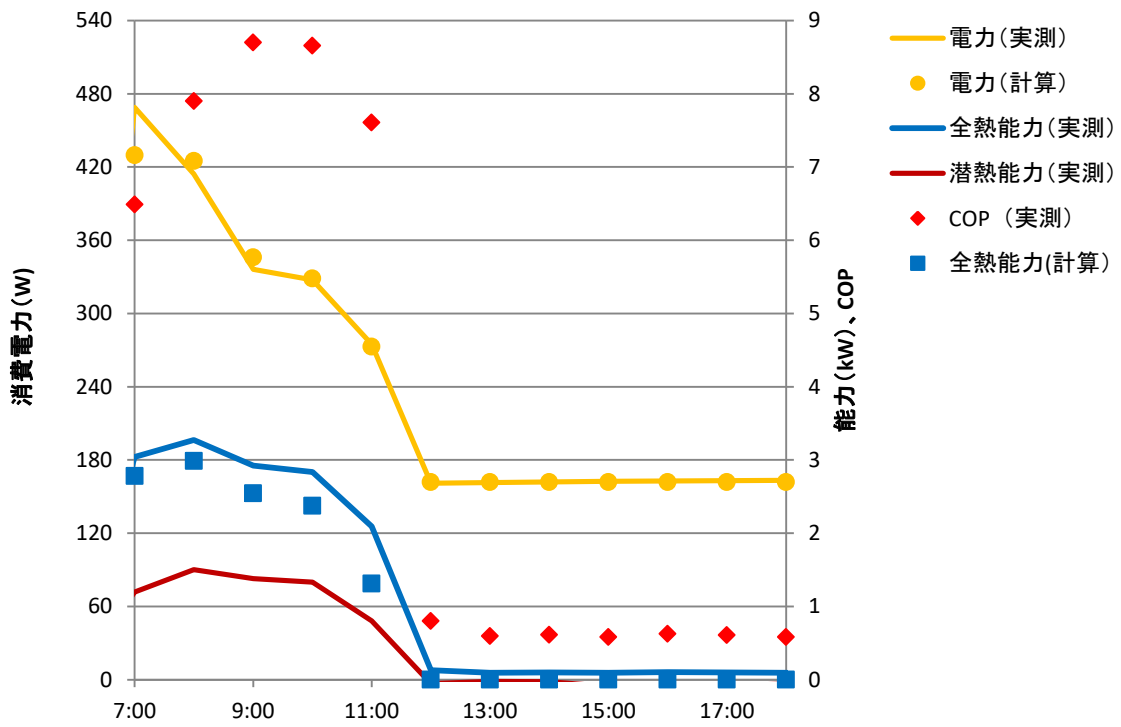


図 197 3/13 運転状況

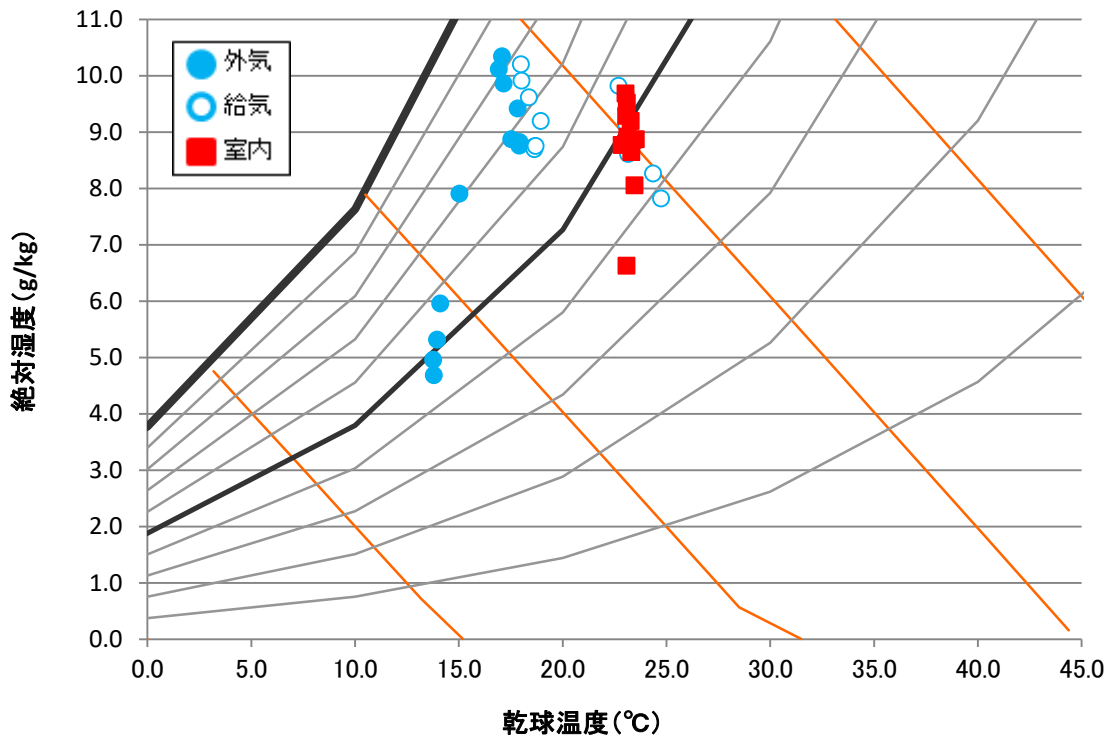


図 198 3/13 空気状況

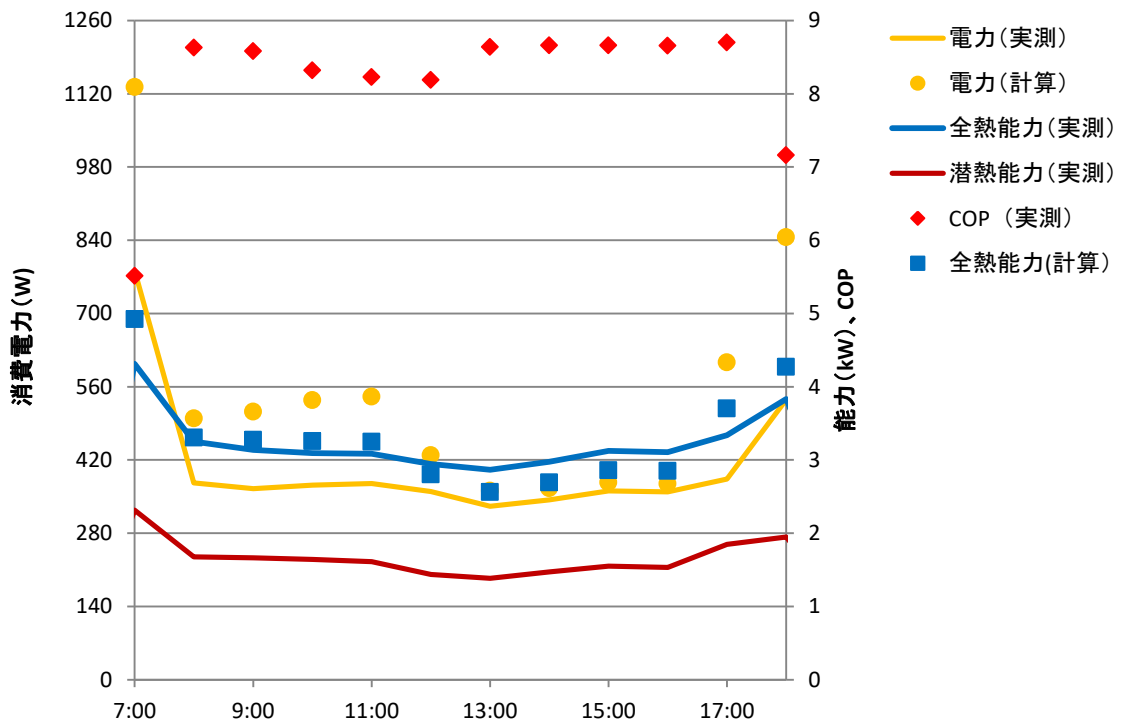


図 199 3/14 運転状況

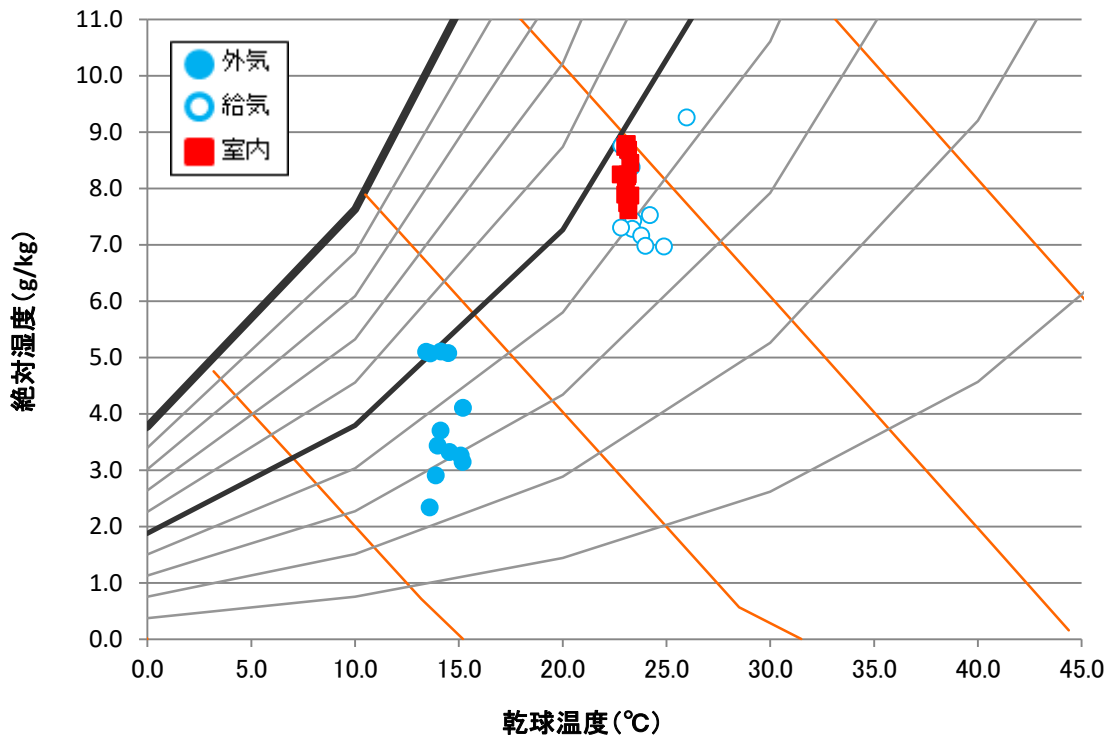


図 200 3/14 空気状況

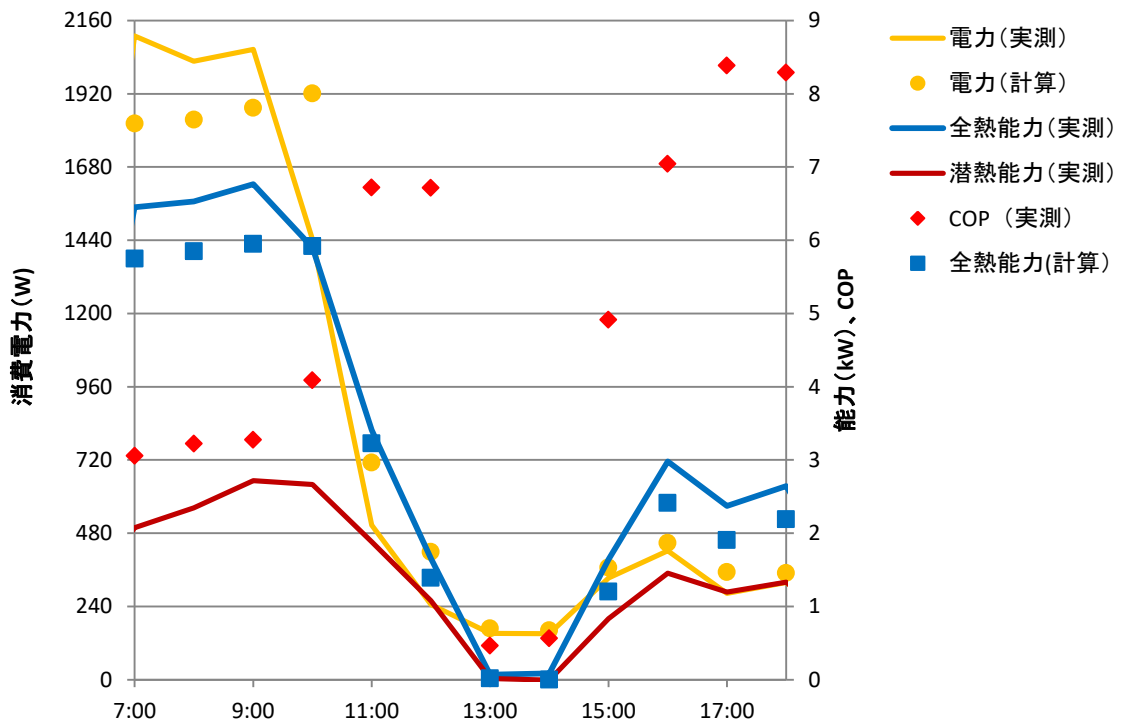


図 201 3/17 運転状況

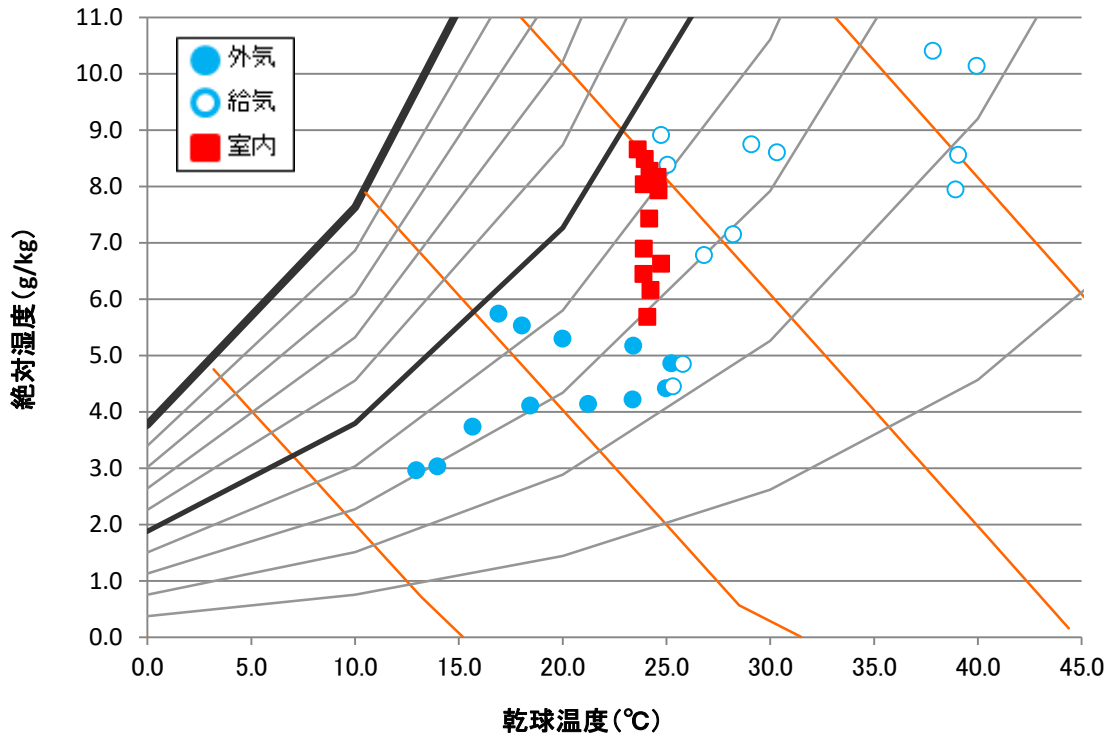


図 202 3/17 空気状況

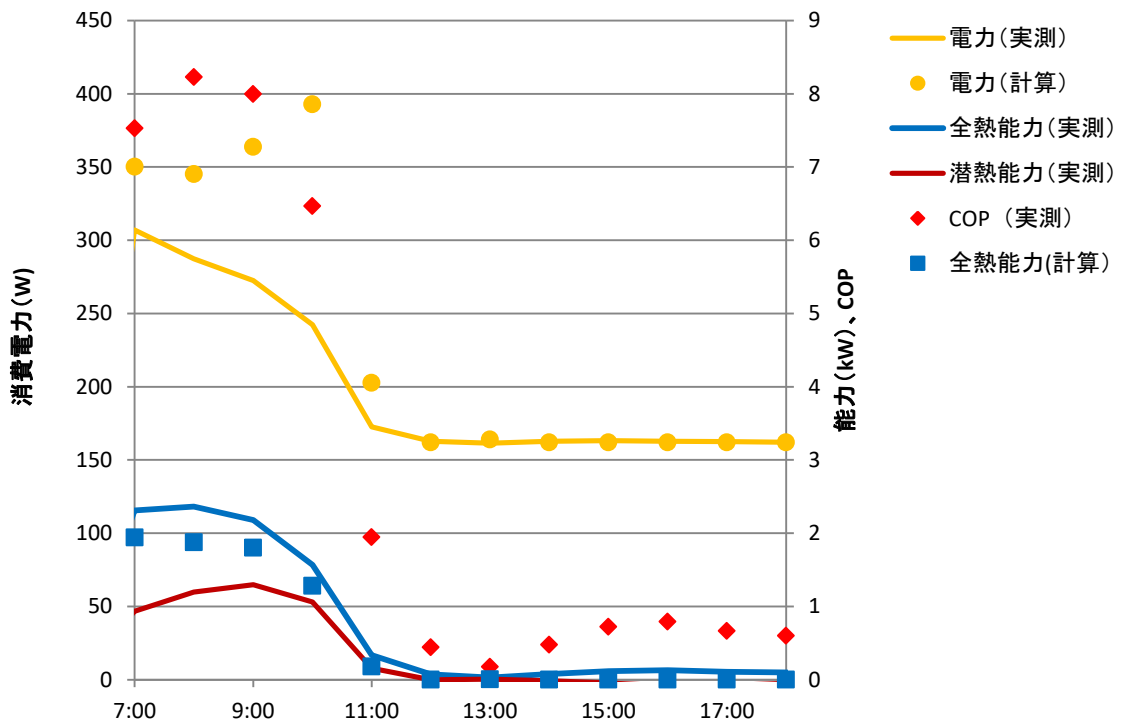


図 203 3/18 運転状況

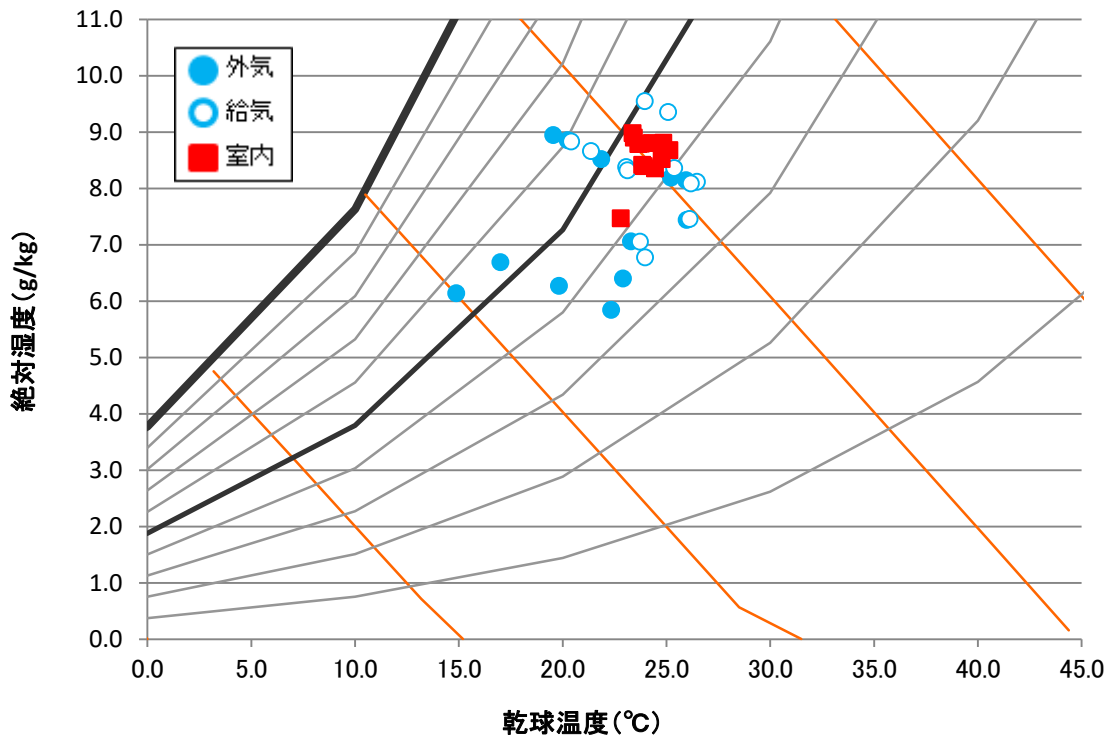


図 204 3/18 空気状況

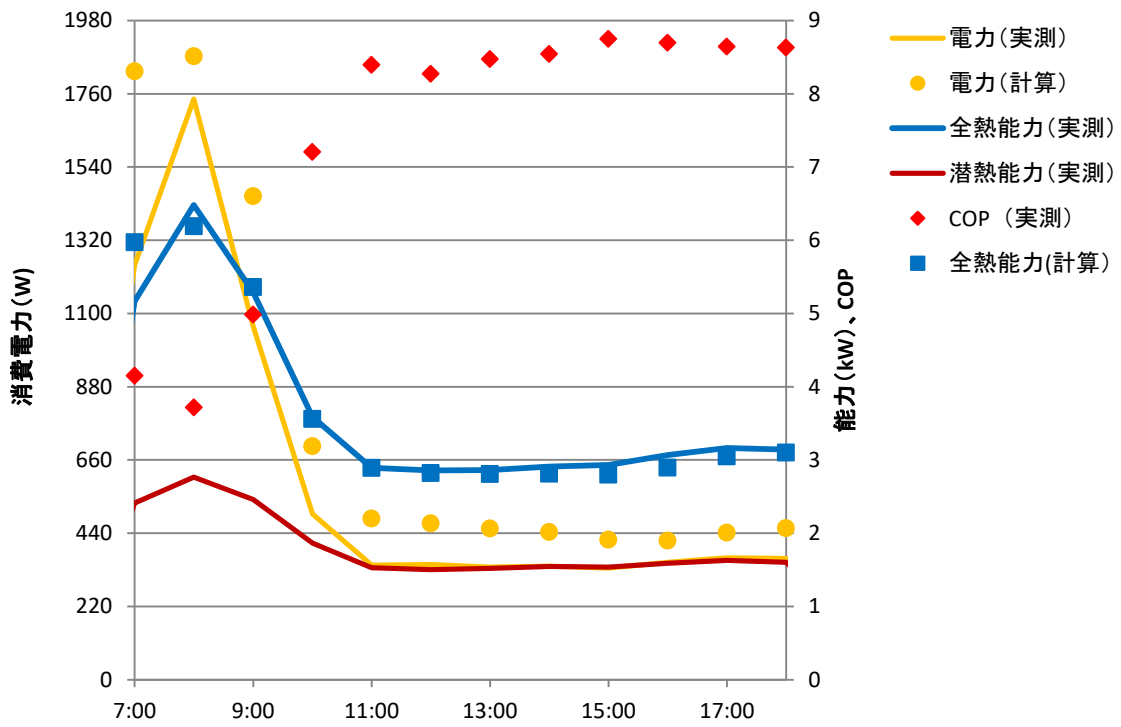


図 205 3/19 運転状況

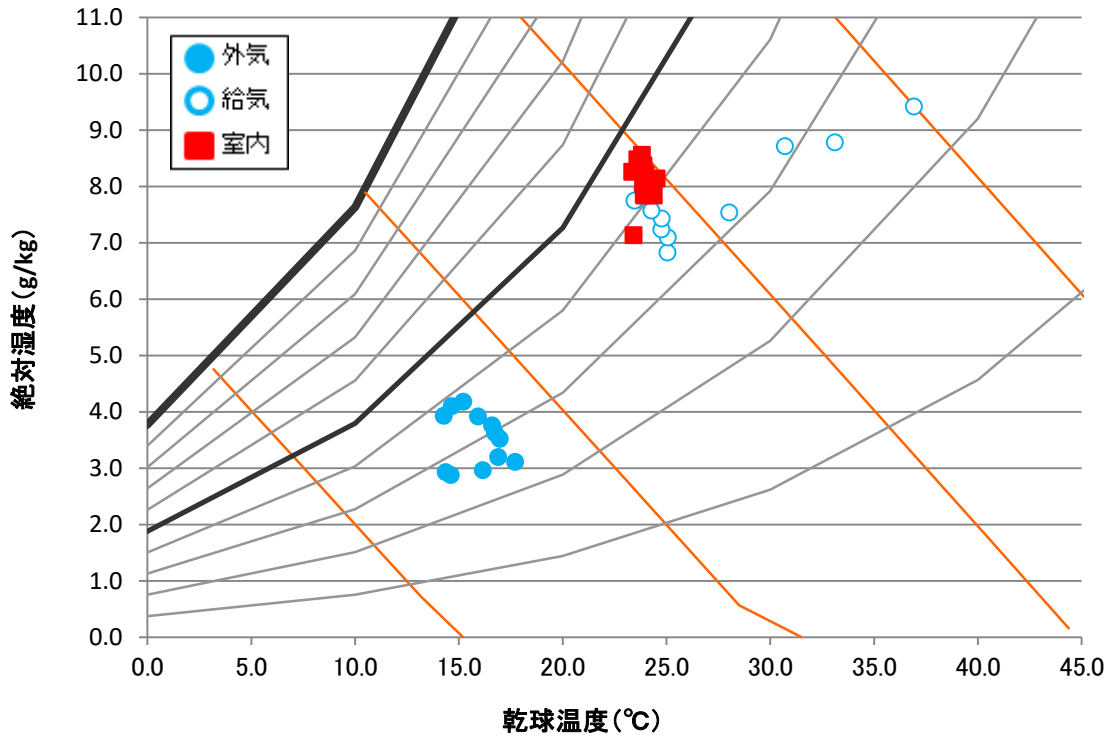


図 206 3/19 空気状況

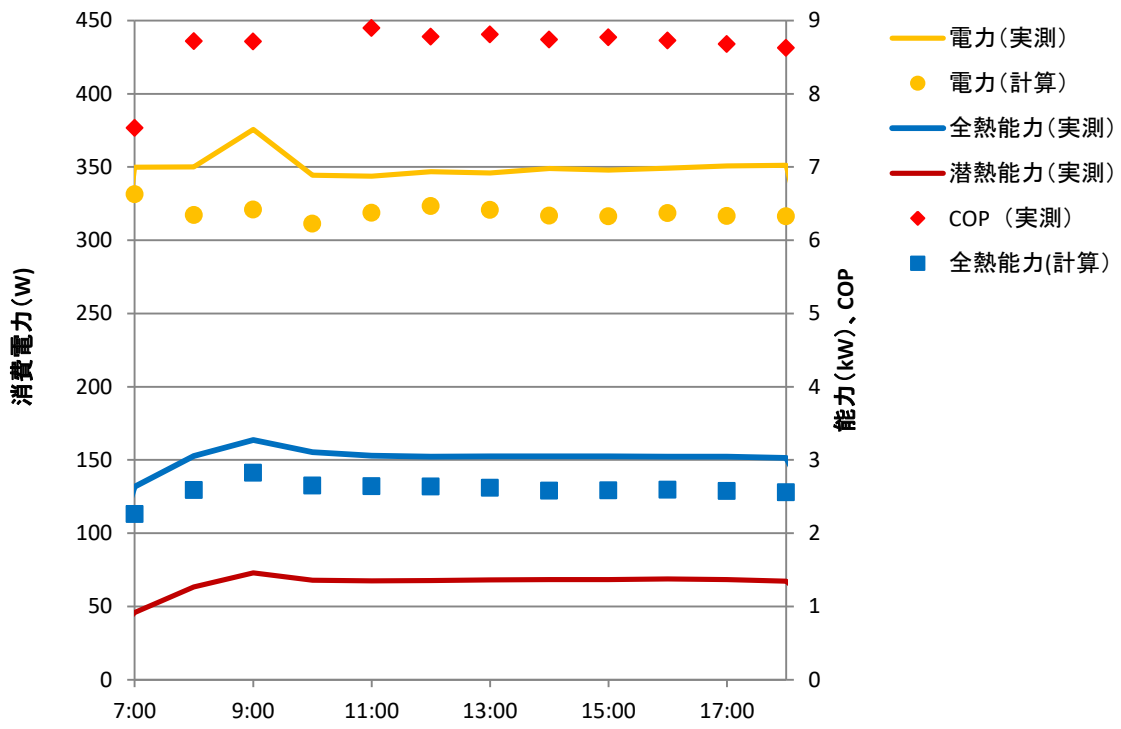


図 207 3/20 運転状況

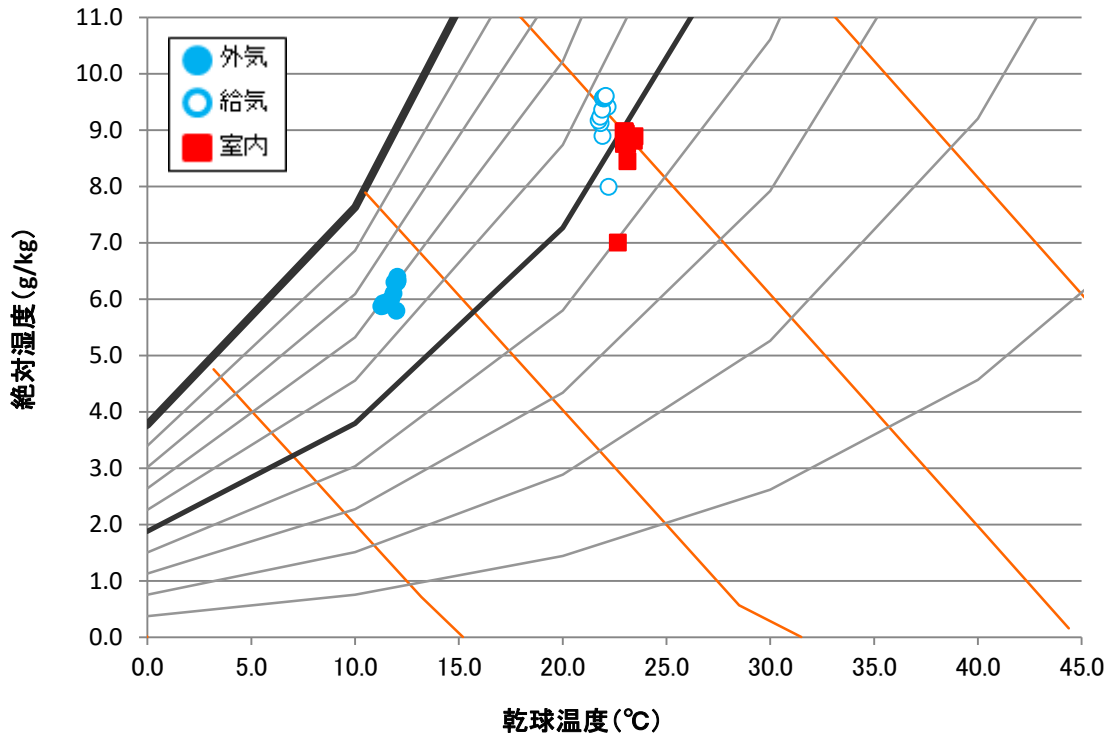


図 208 3/20 空気状況

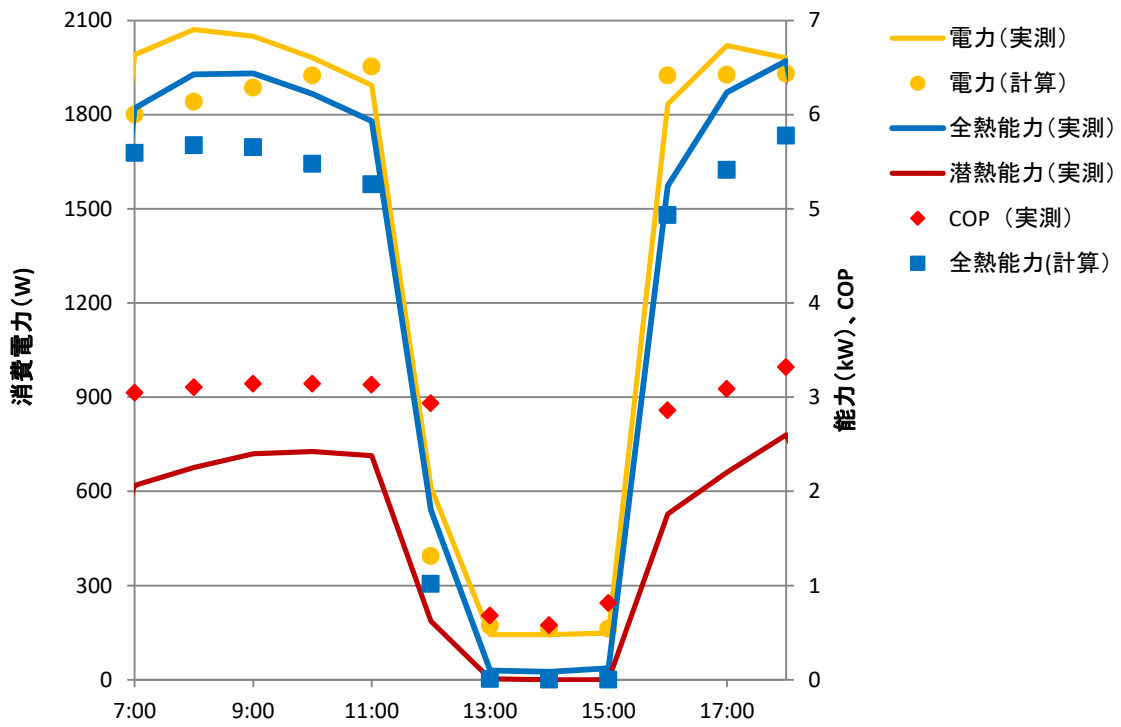


図 209 3/24 運転状況

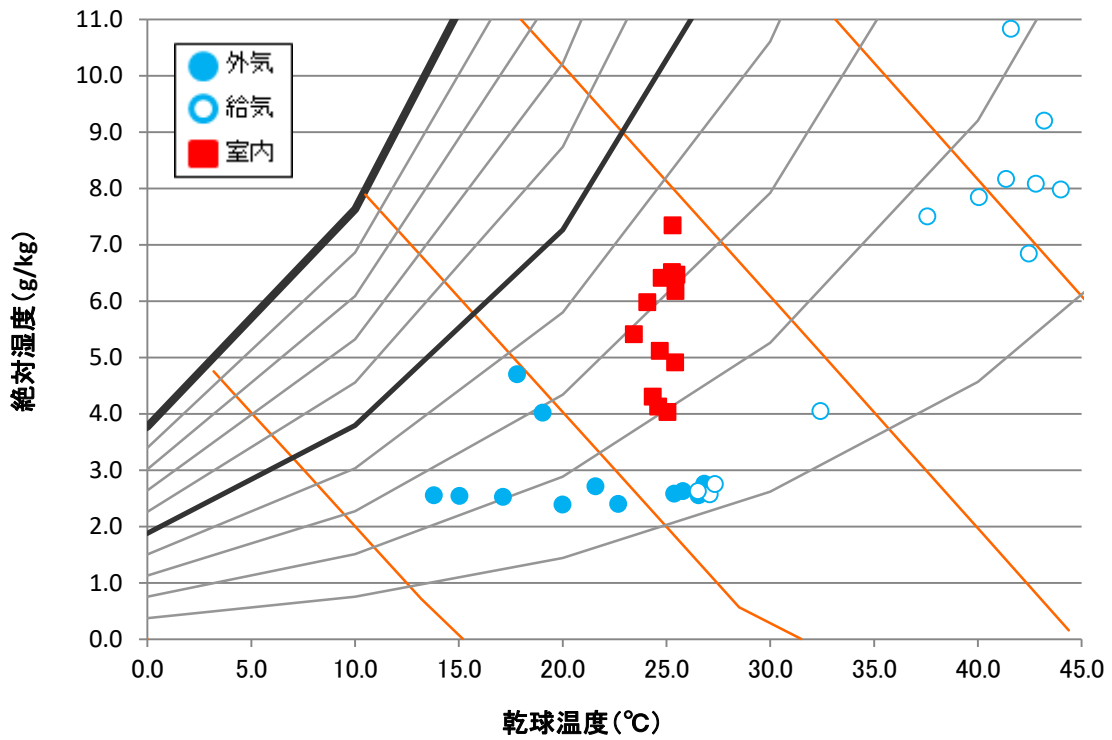


図 210 3/24 空気状況

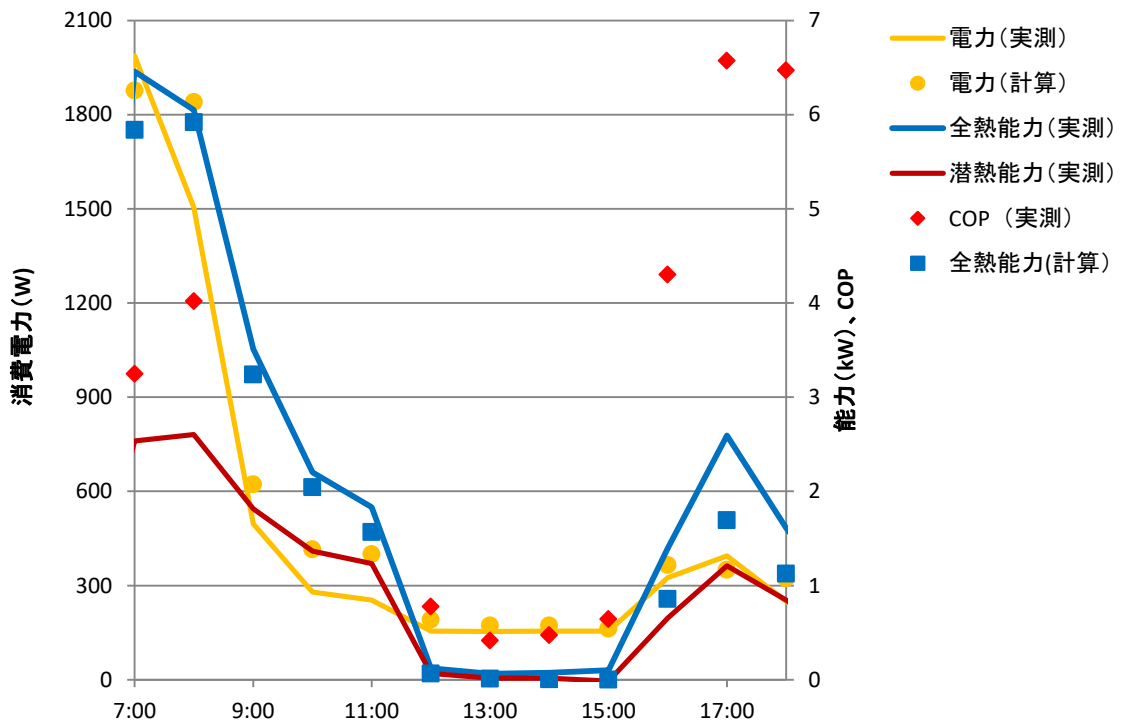


図 211 3/25 運転状況

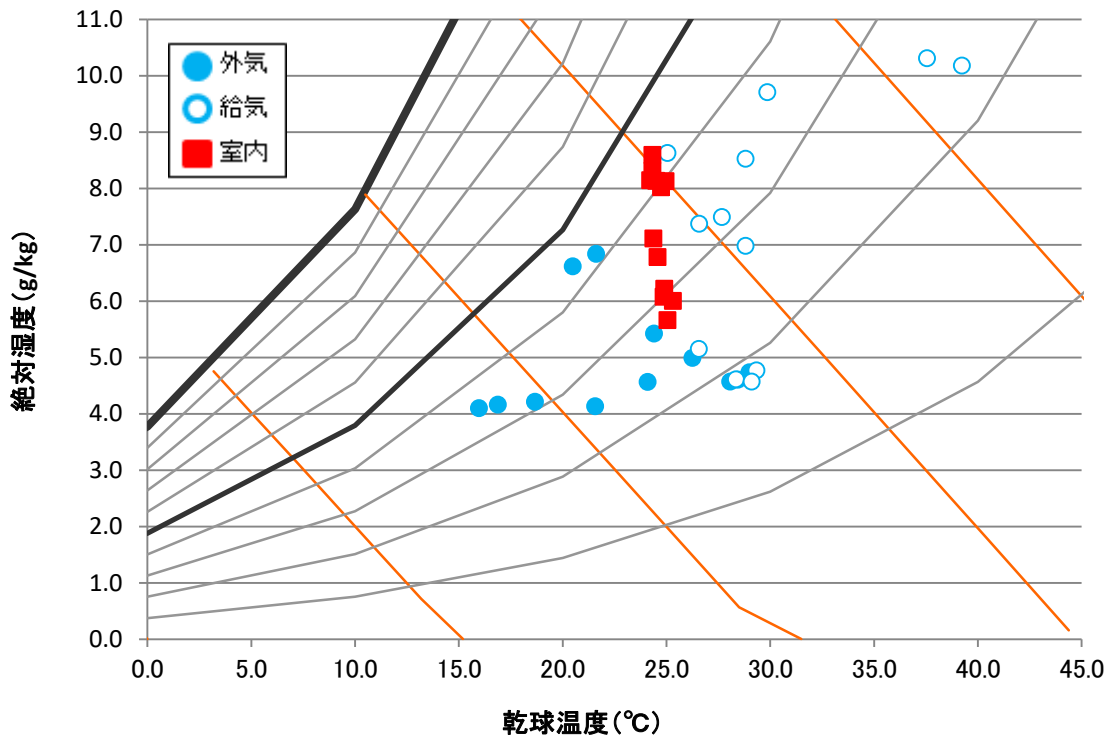


図 212 3/25 空気状況

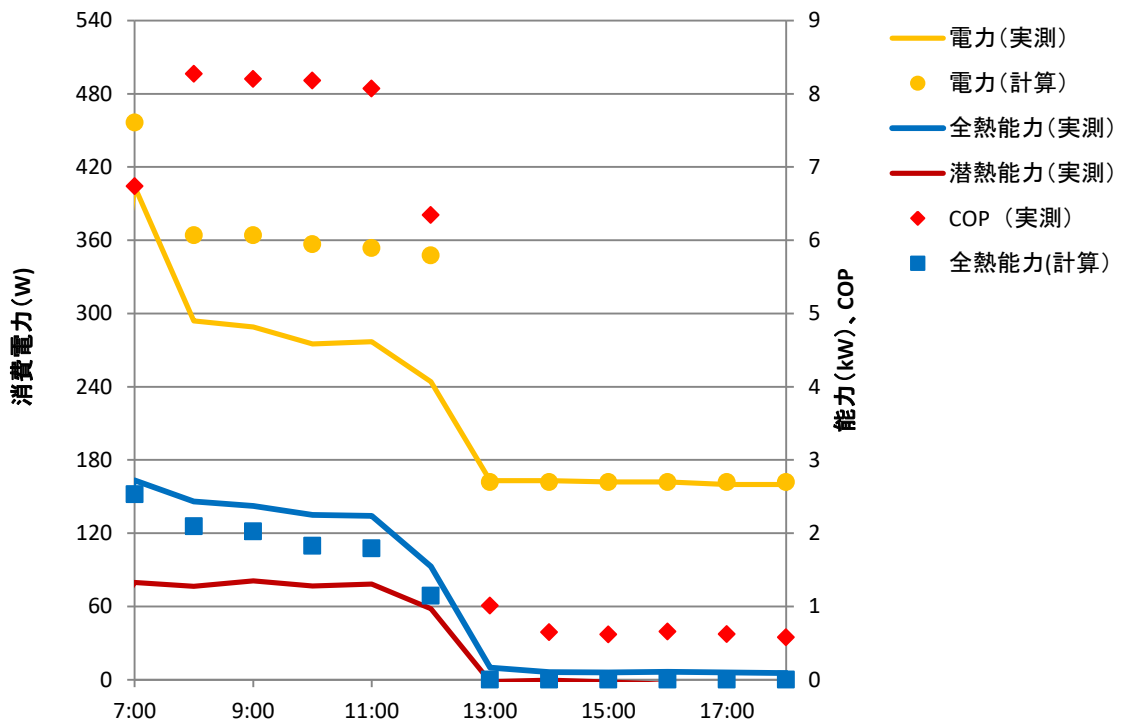


図 213 3/26 運転状況

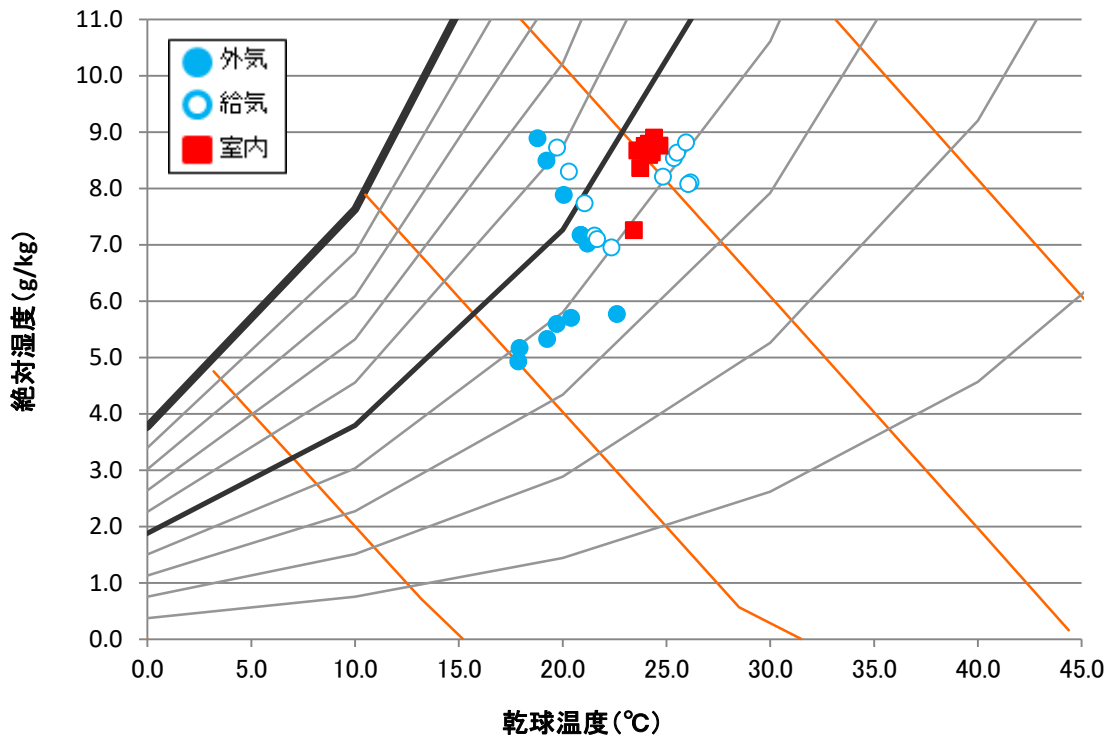


図 214 3/26 空気状況

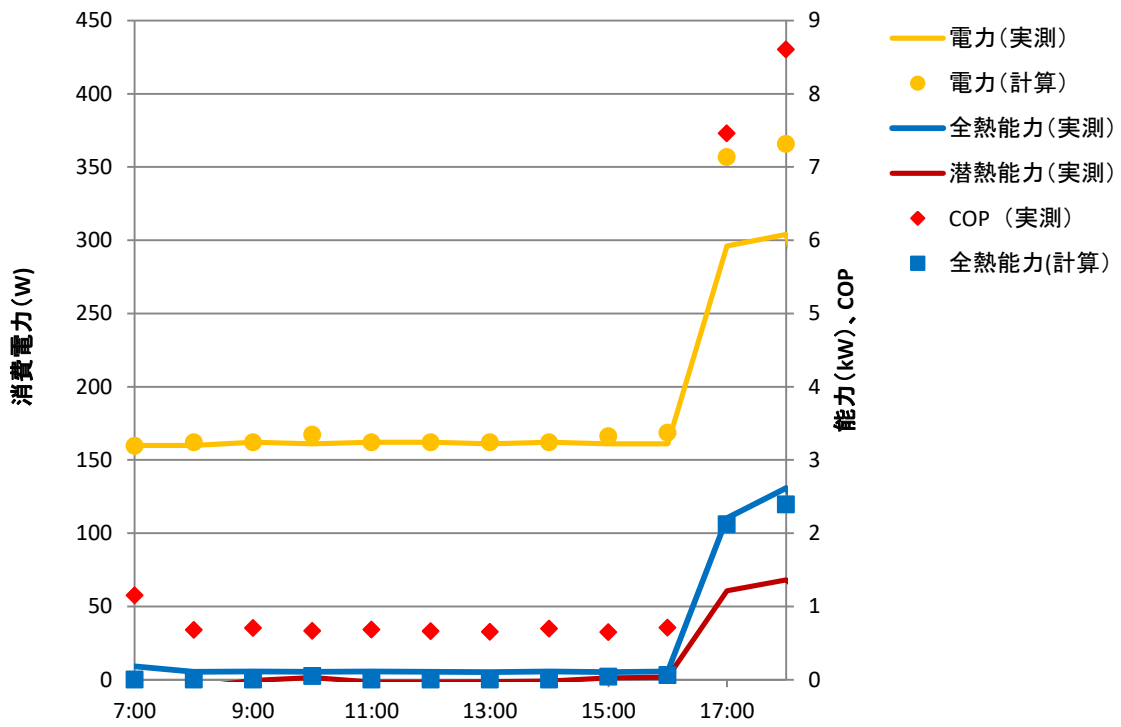


図 215 3/27 運転状況

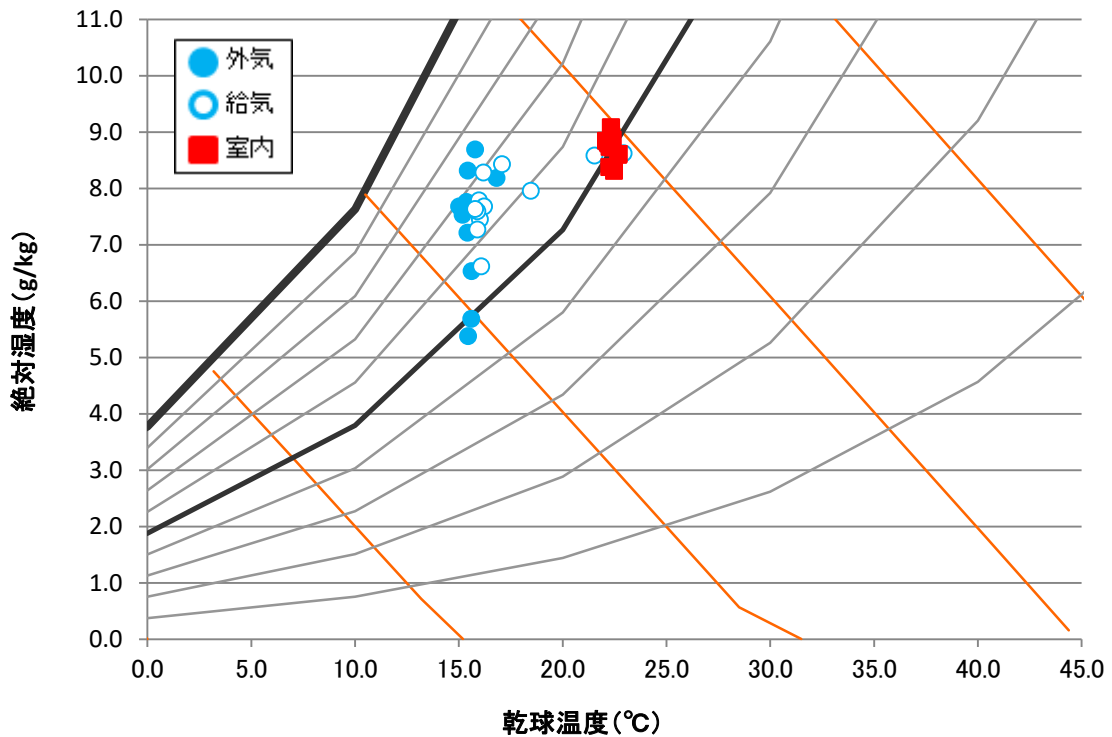


図 216 3/27 空気状況

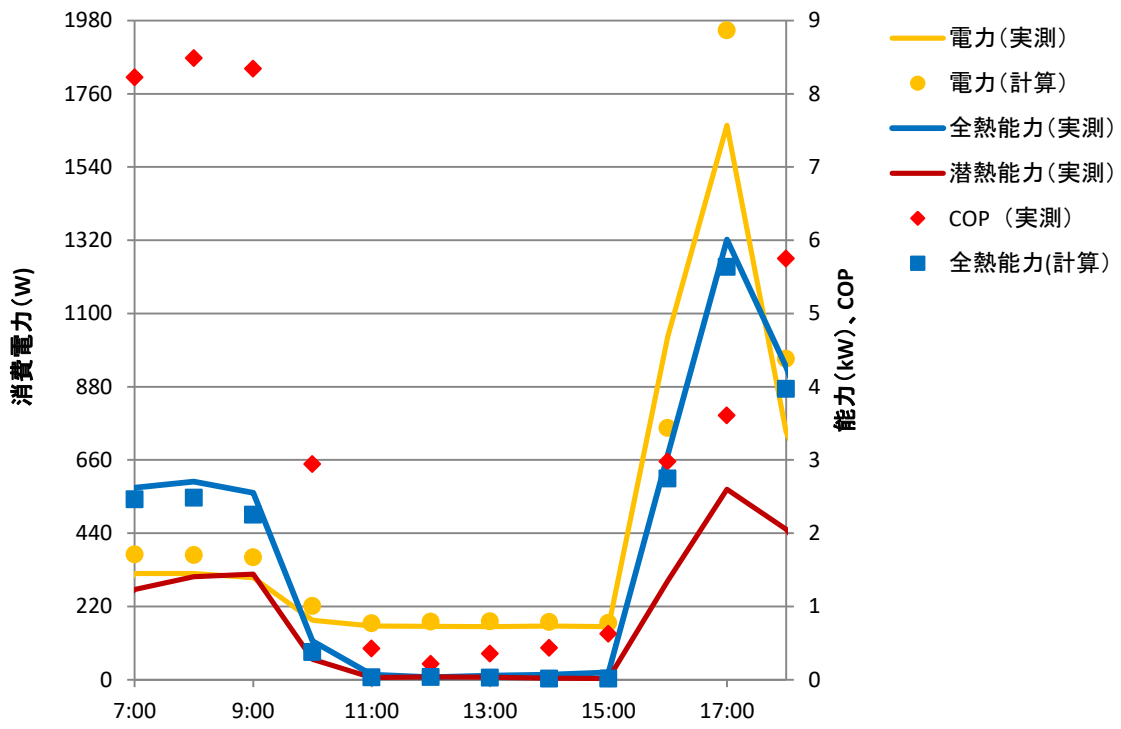


図 217 3/28 運転状況

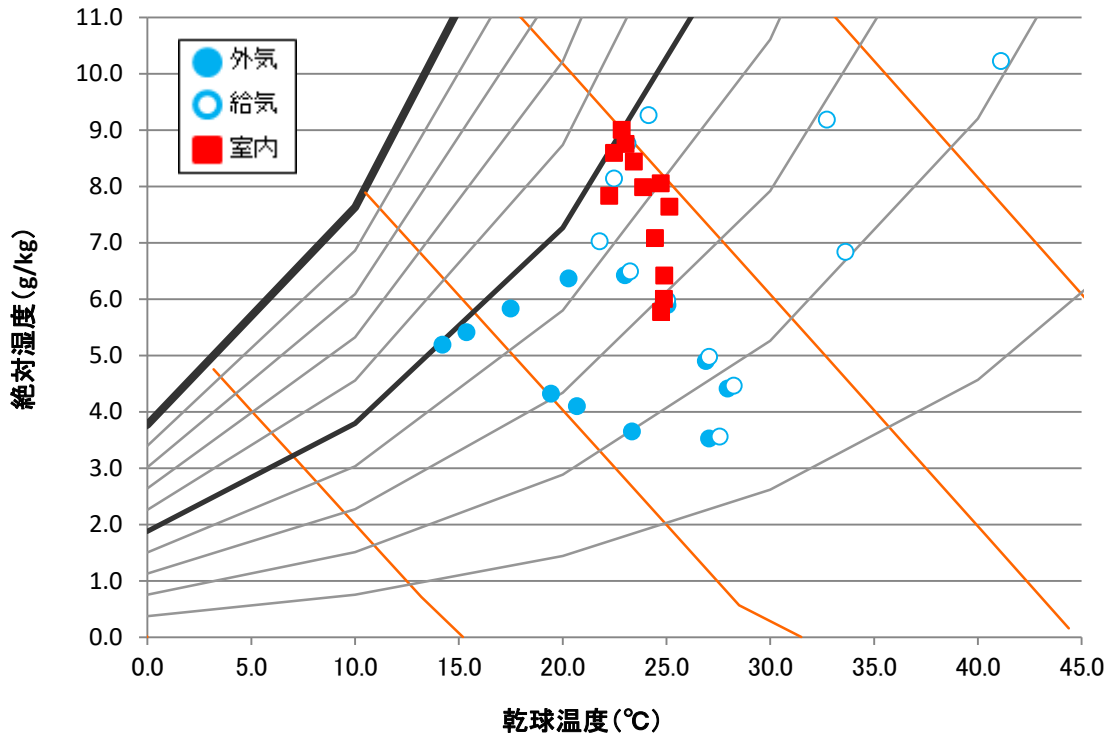


図 218 3/28 空気状況

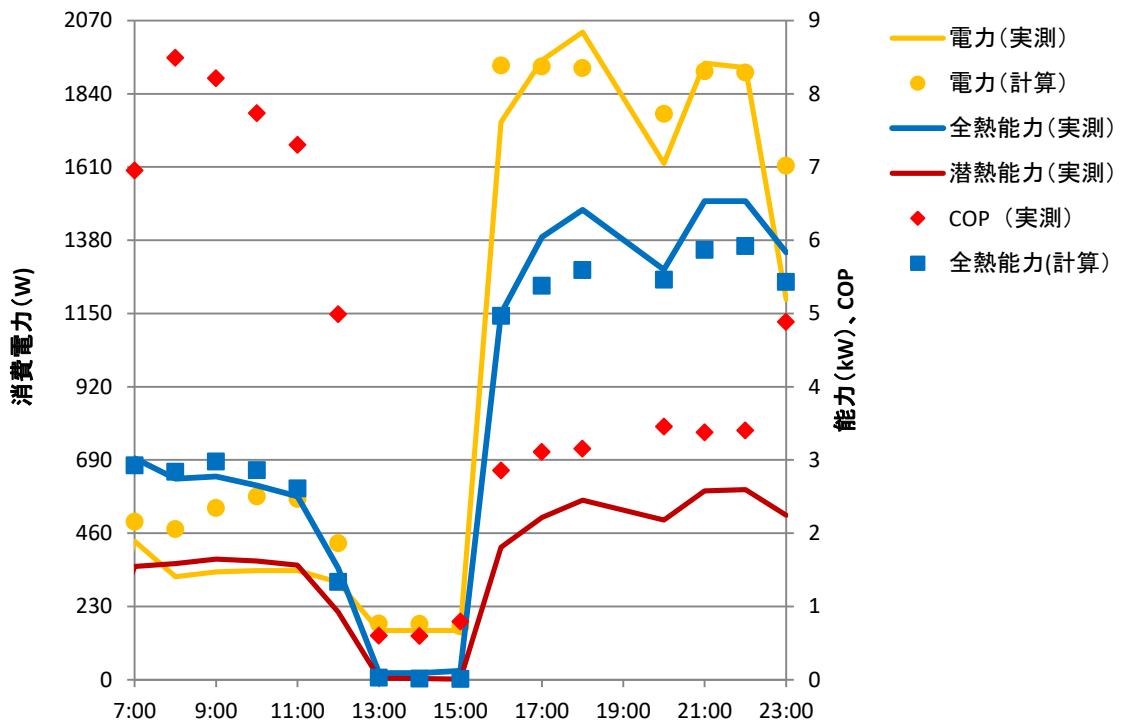


図 219 3/31 運転状況

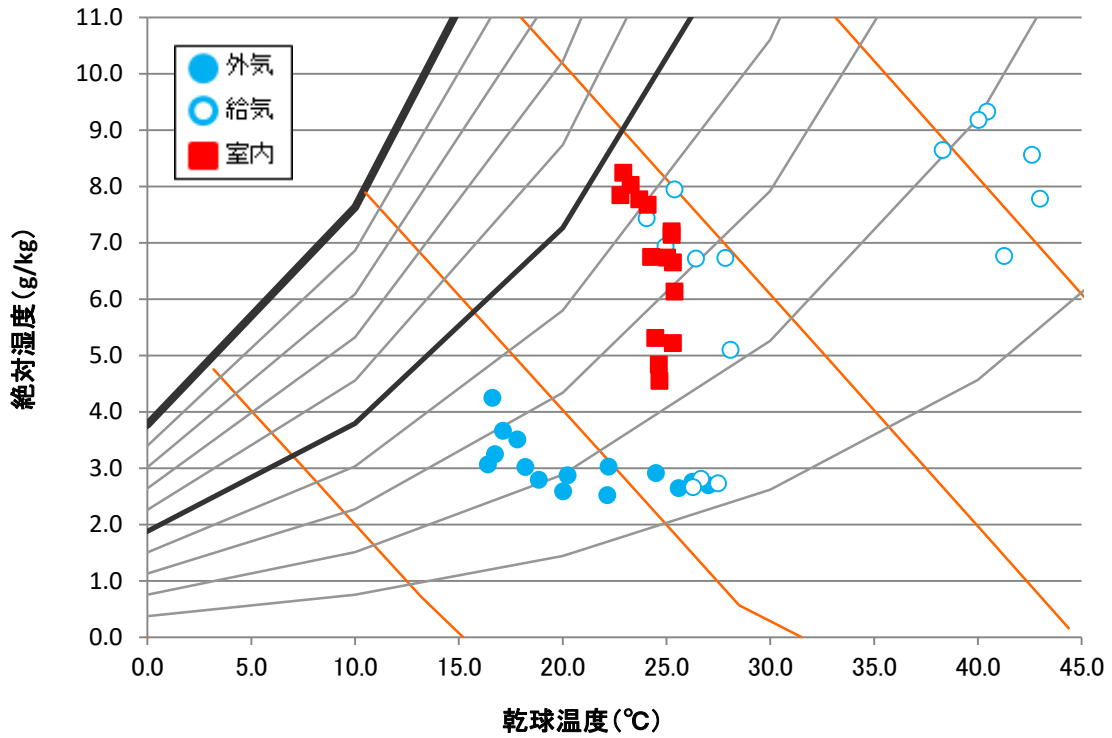


図 220 3/31 空気状況

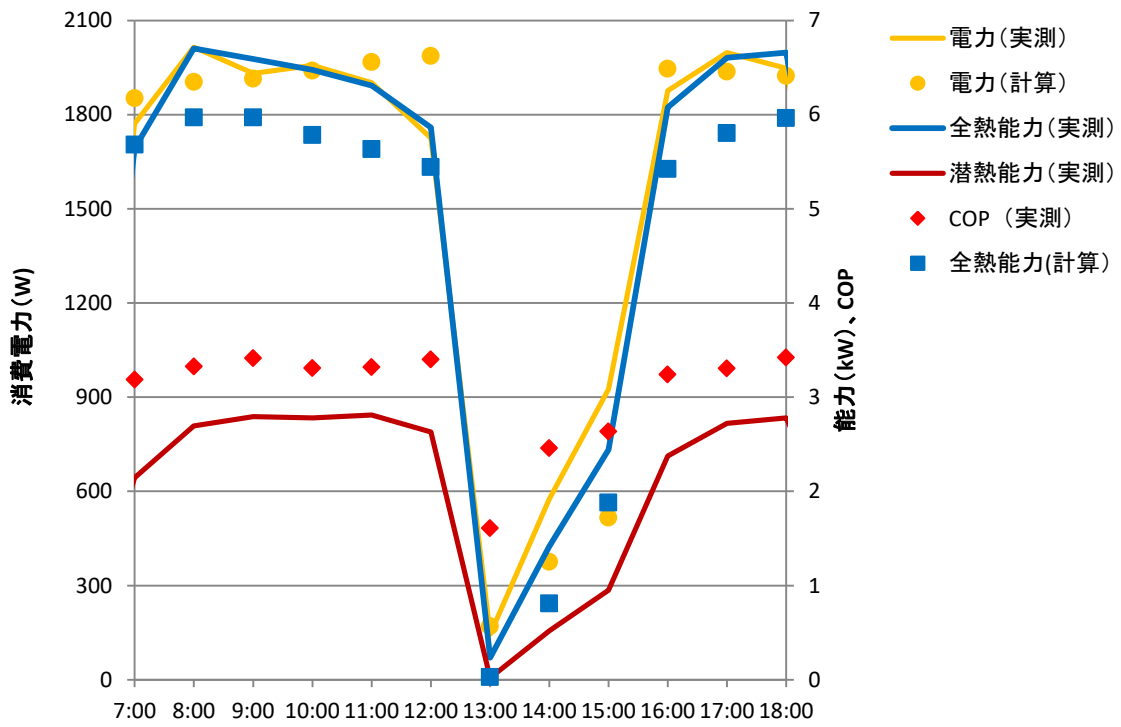


図 221 4/1 運転状況

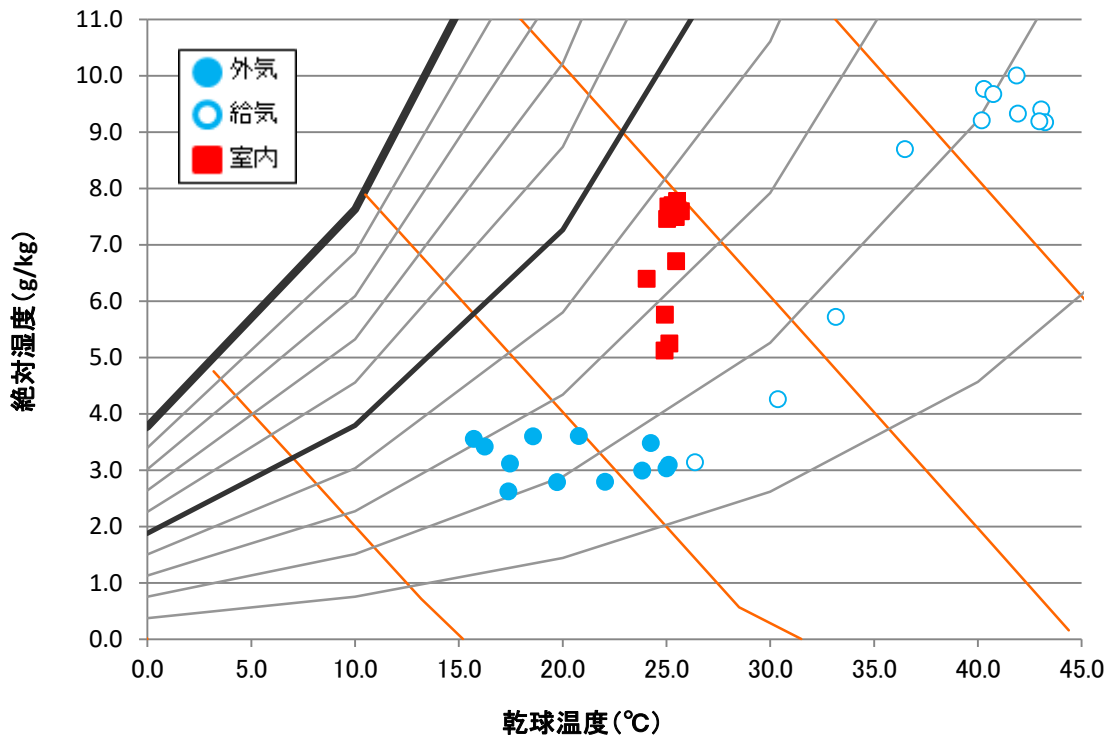


図 222 4/1 空気状況

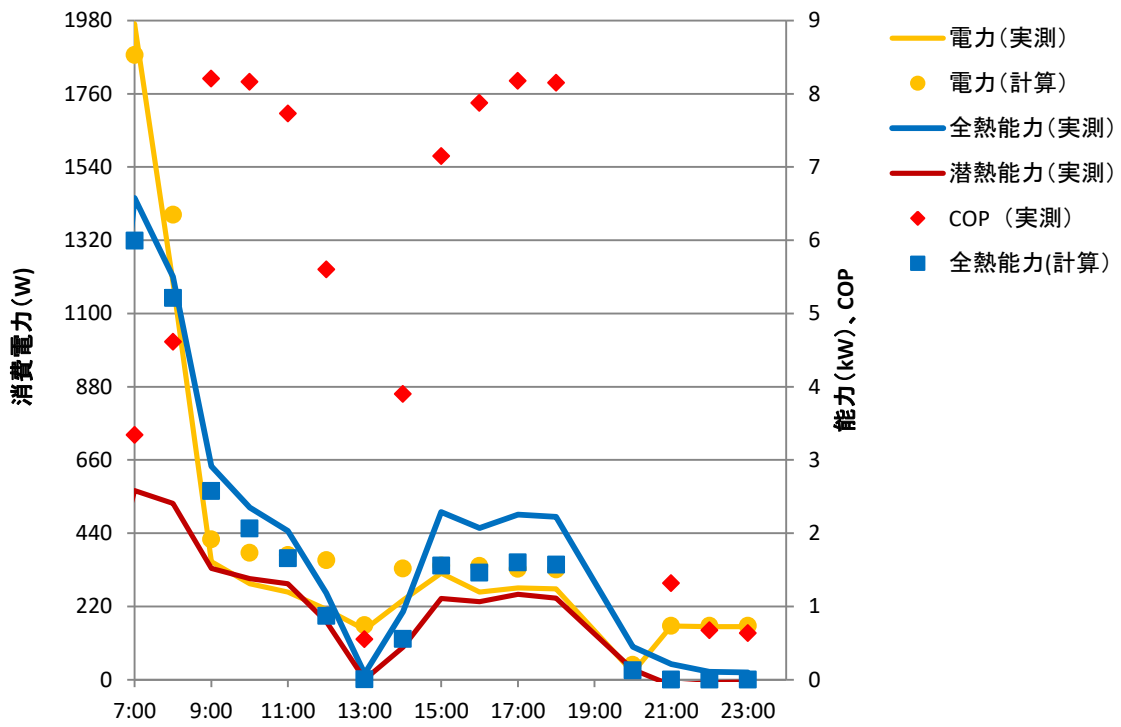


図 223 4/2 運転状況

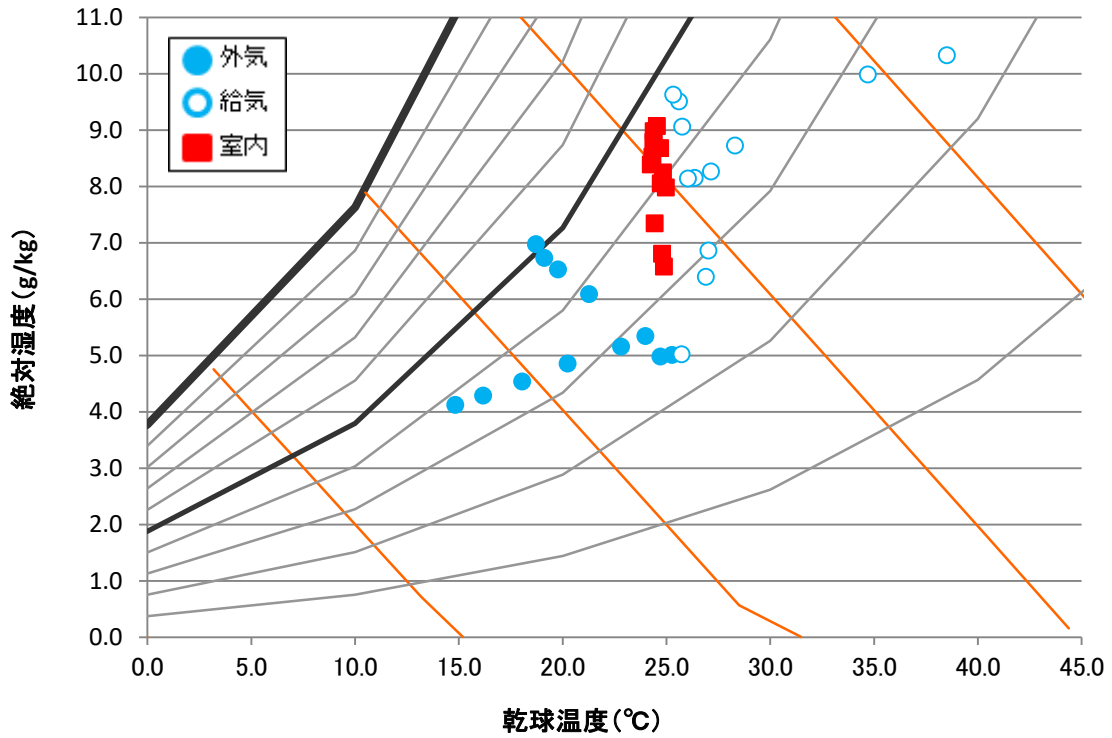


図 224 4/2 空気状況

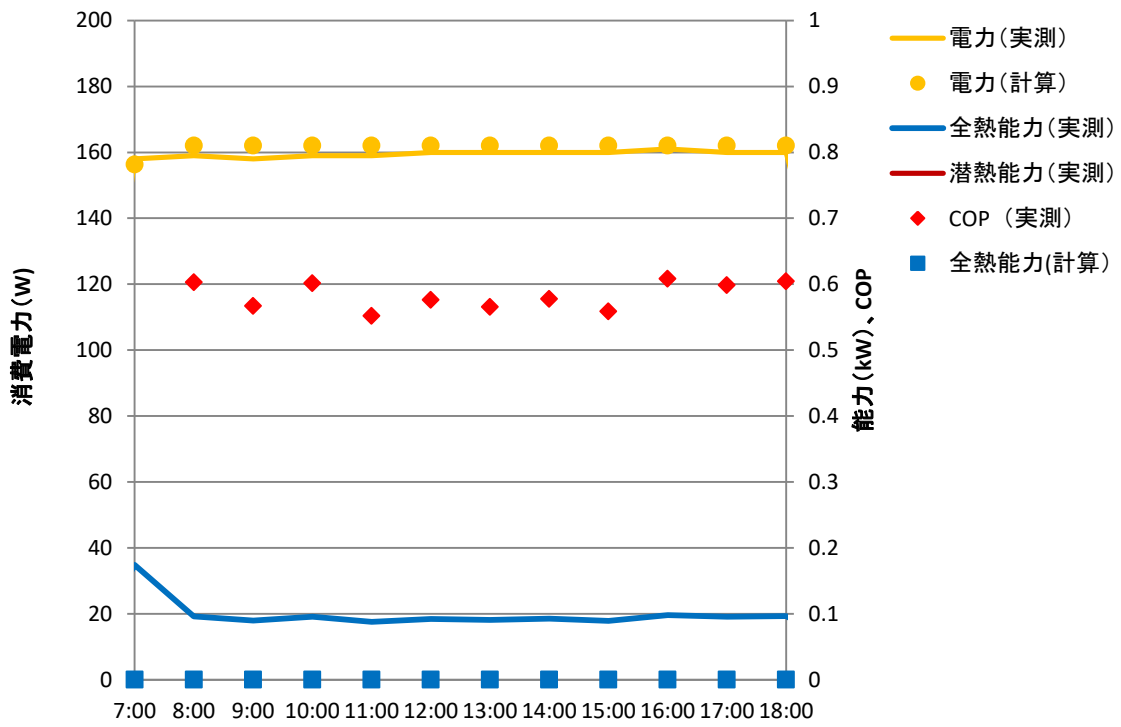


図 225 4/3 運転状況

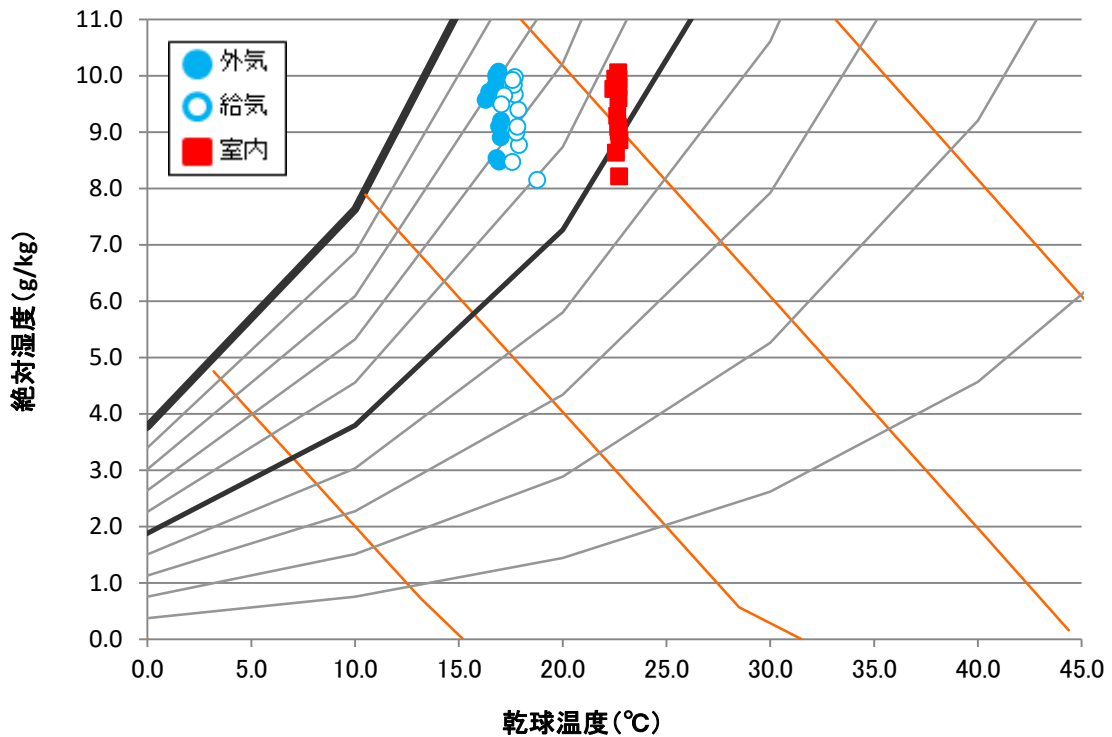


図 226 4/3 空気状況

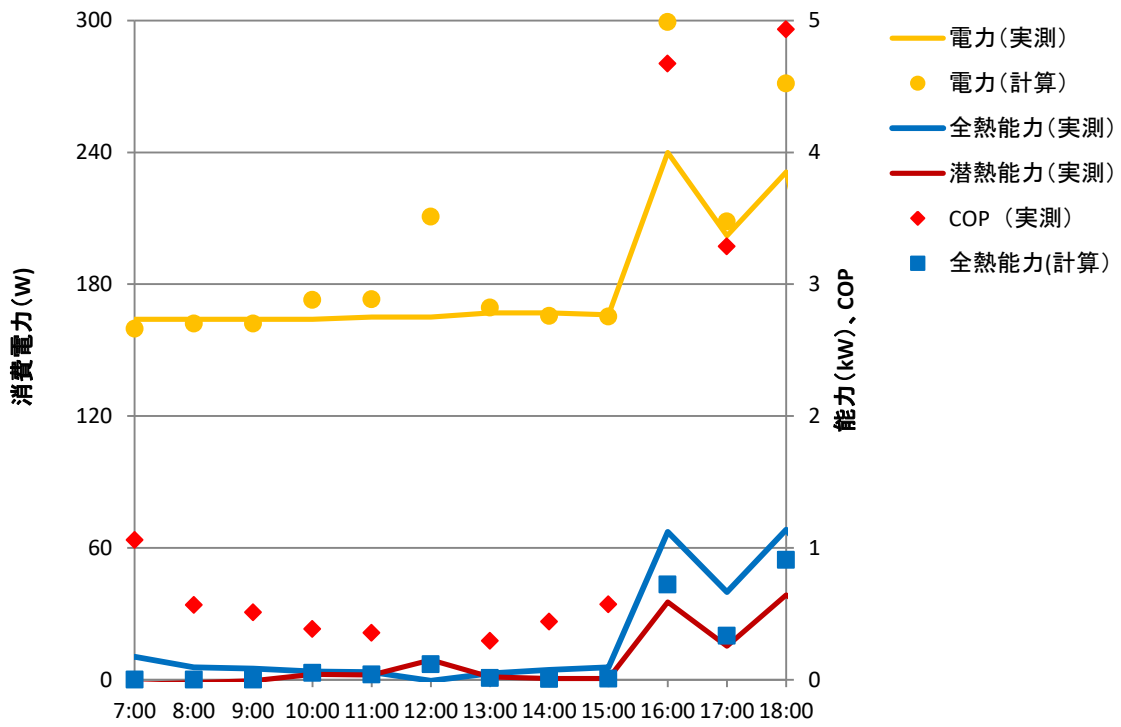


図 227 4/4 運転状況

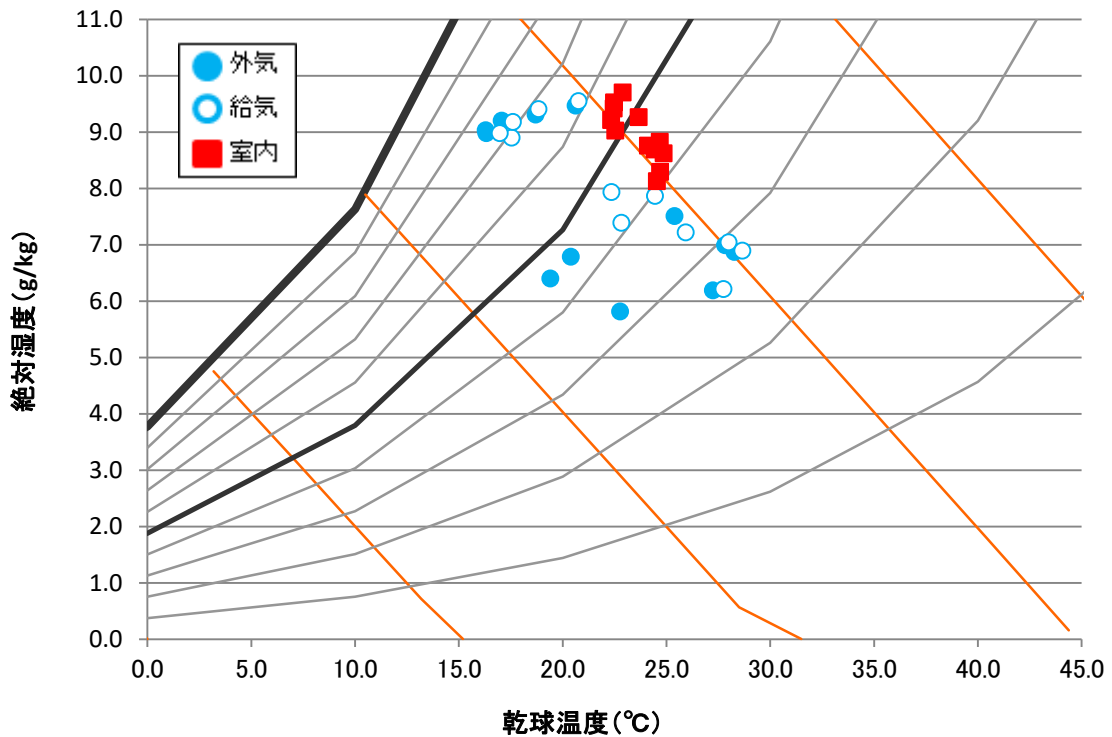


図 228 4/4 空気状況

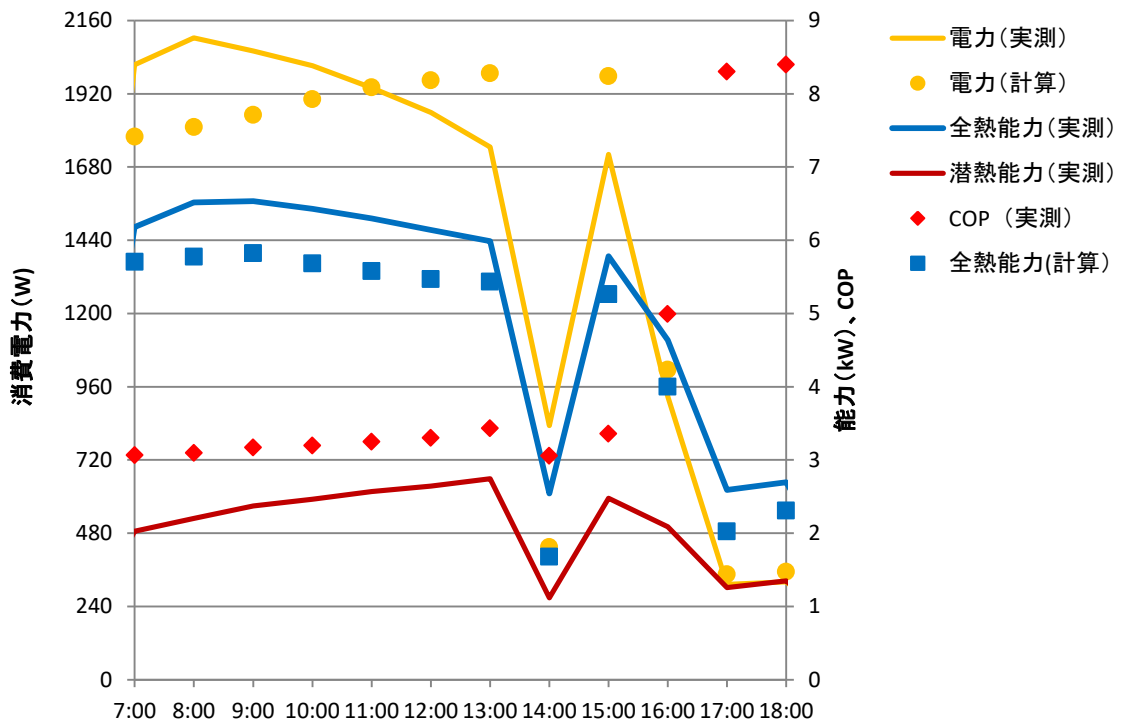


図 229 4/7 運転状況

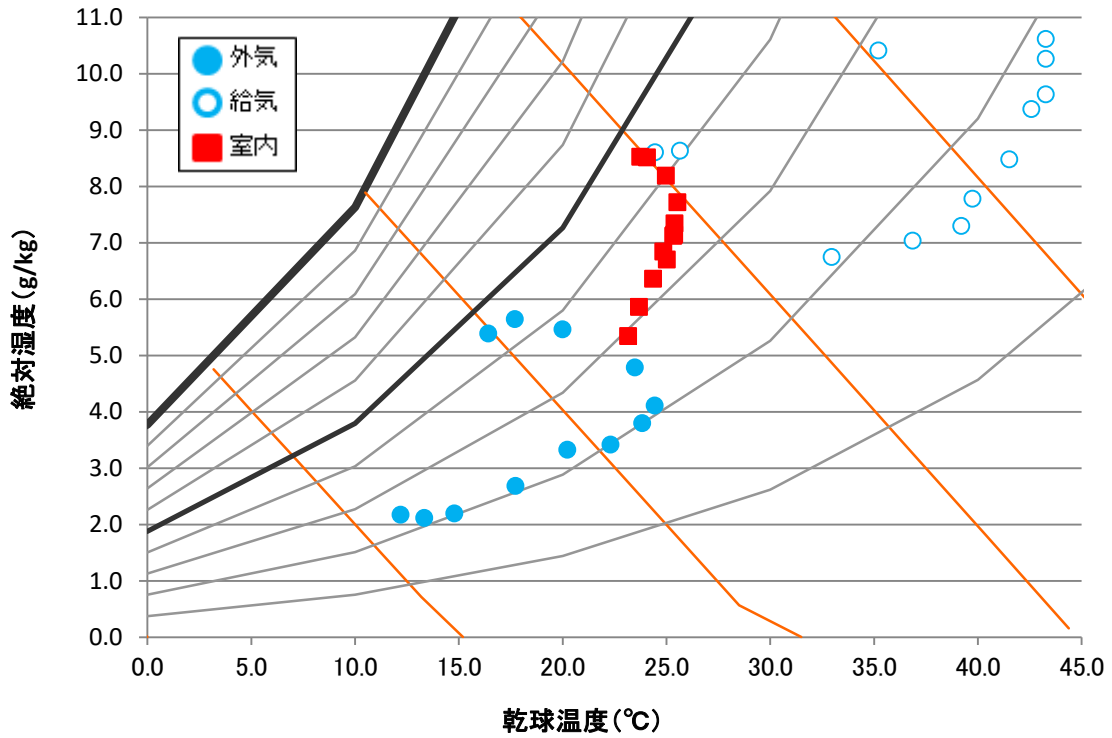


図 230 4/7 空気状況

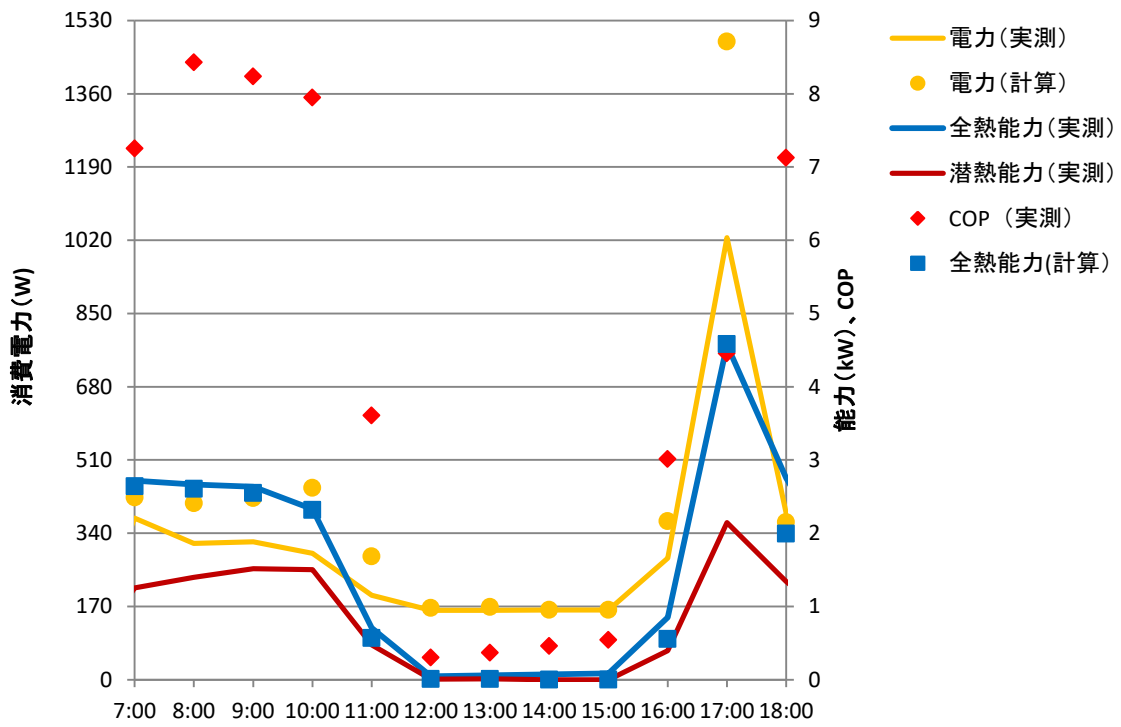


図 231 4/8 運転状況

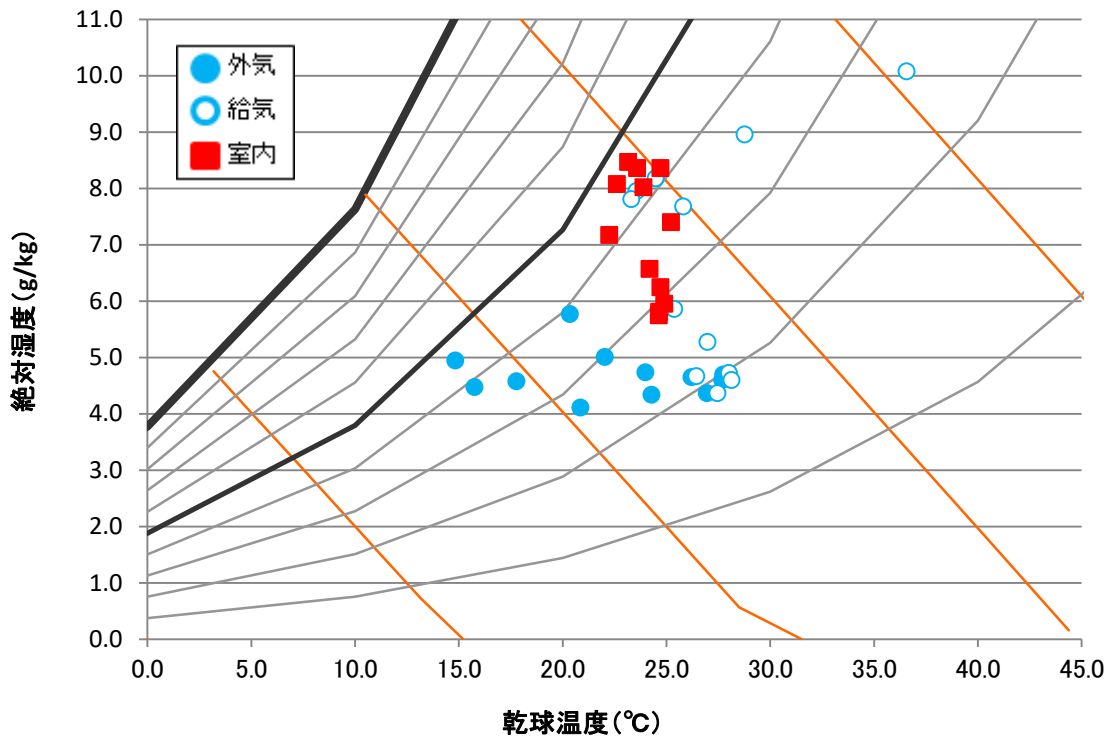


図 232 4/8 空気状況

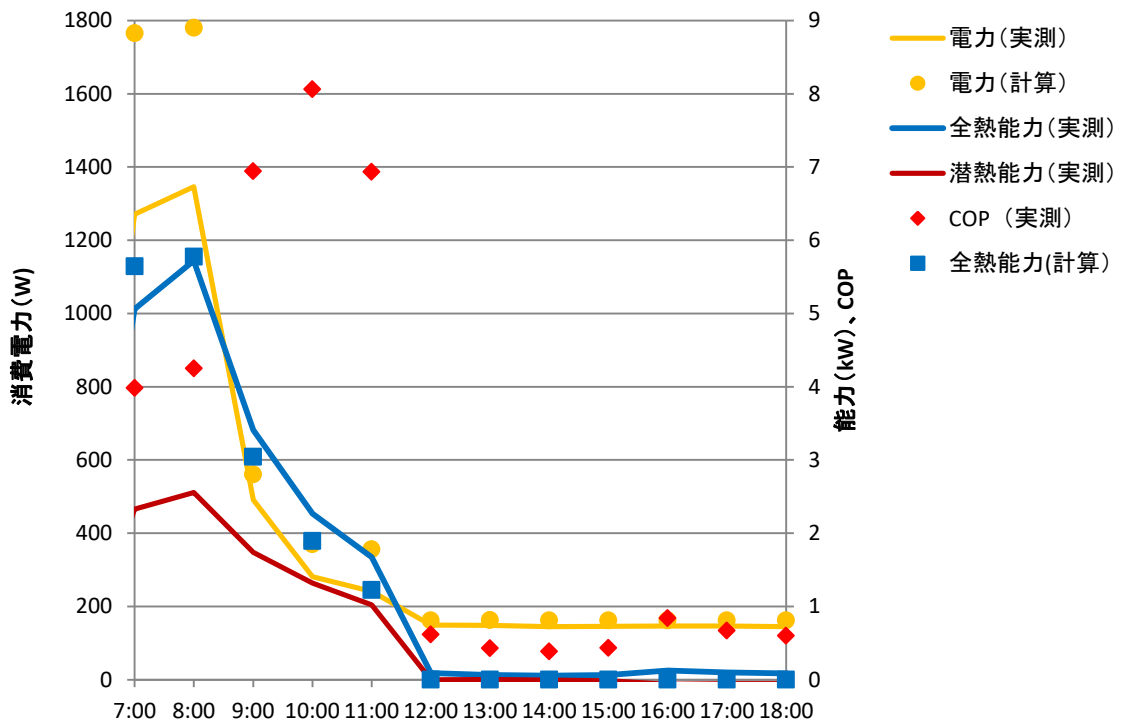


図 233 4/9 運転状態

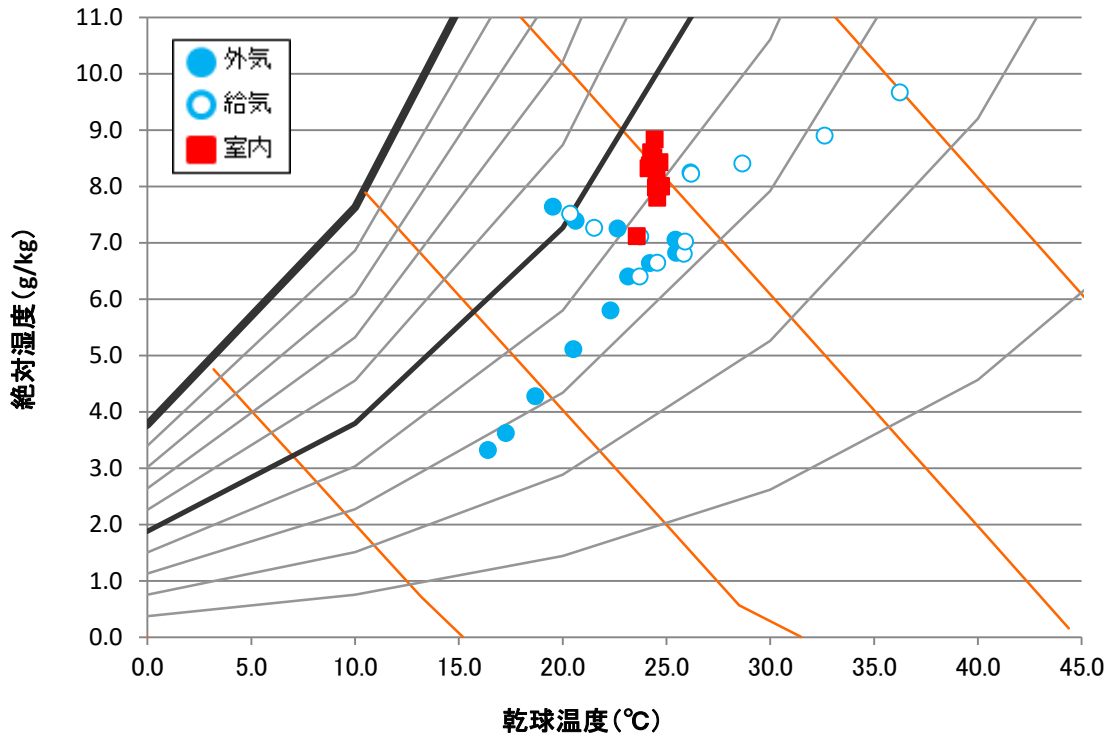


図 234 4/9 空気状態

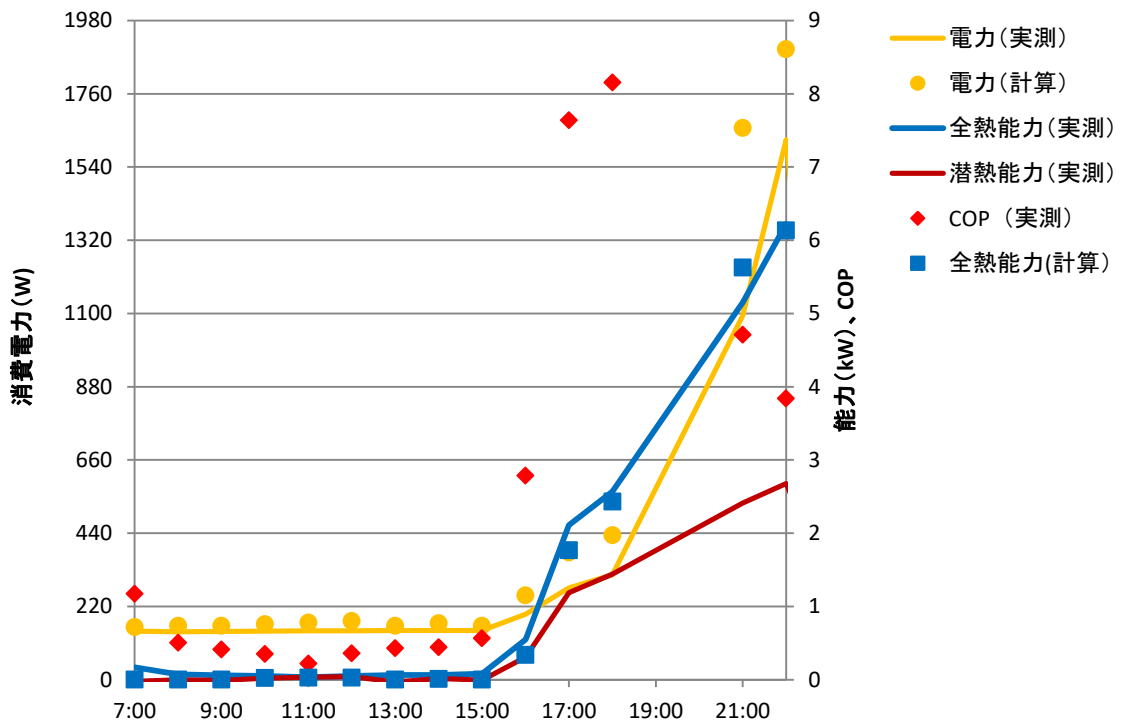


図 235 4/10 運転状況

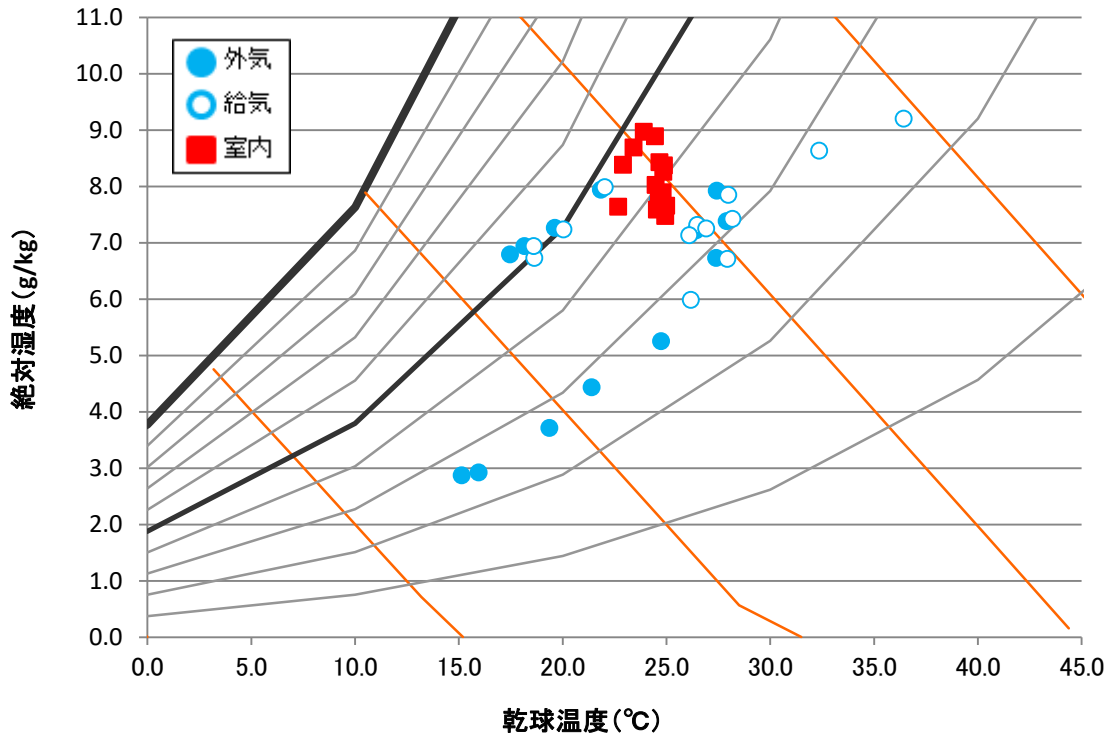


図 236 4/10 空気状況

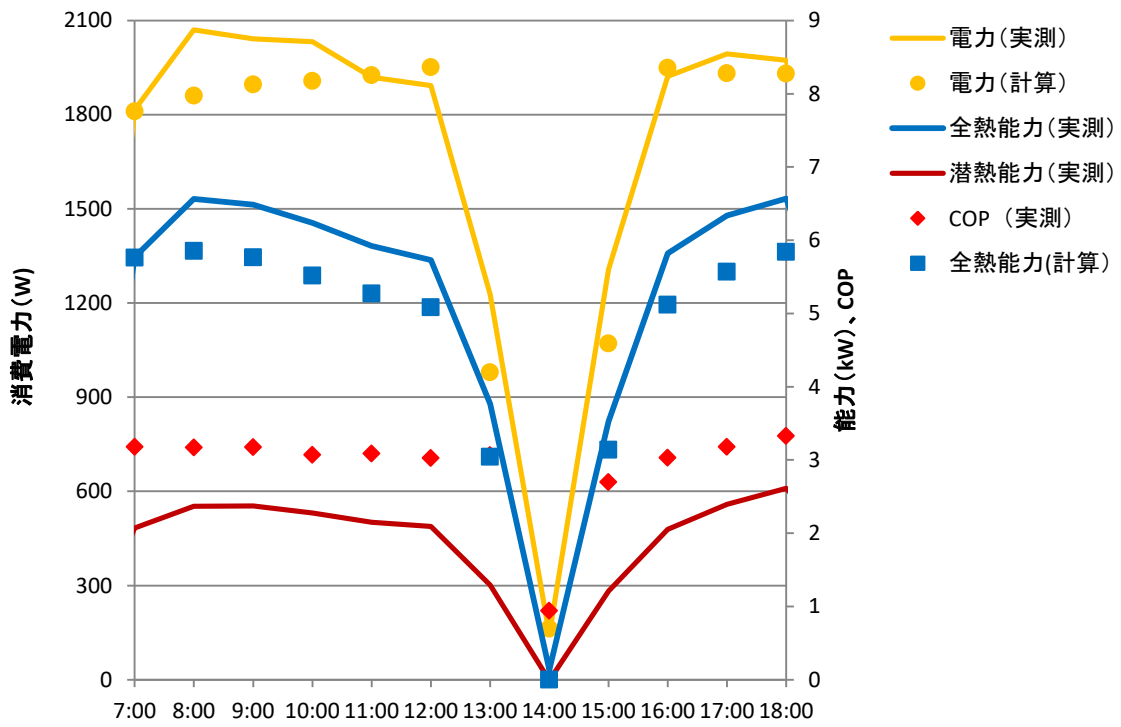


図 237 4/11 運転状況

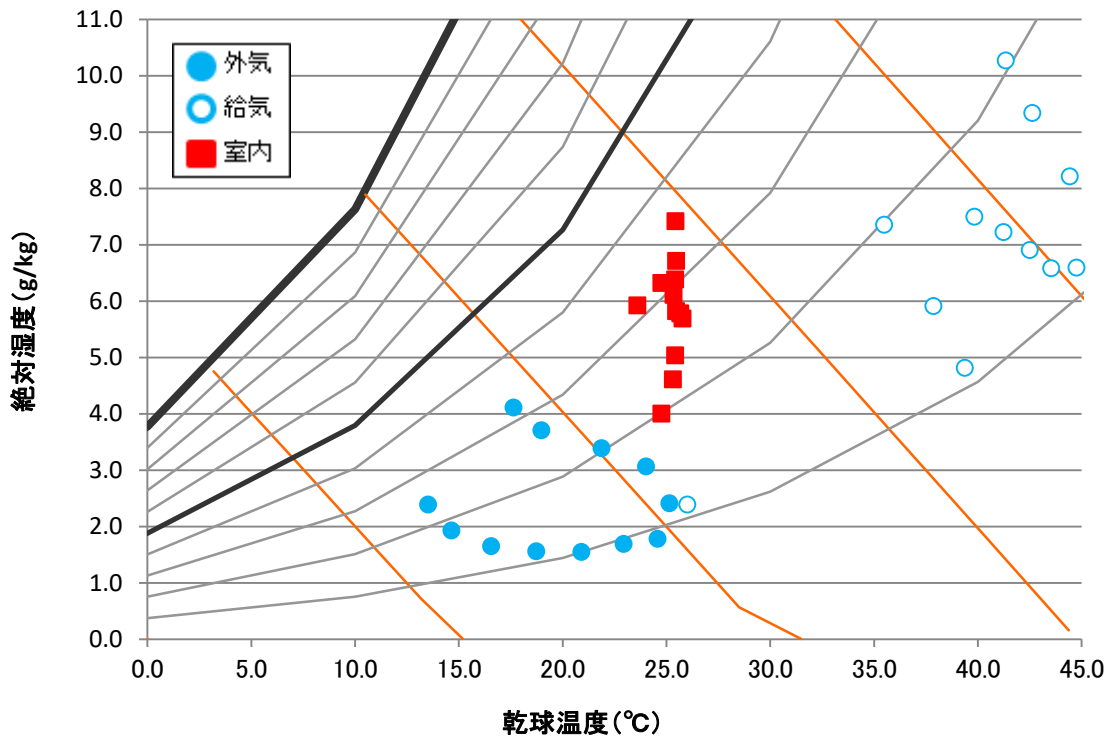


図 238 4/11 空気状況

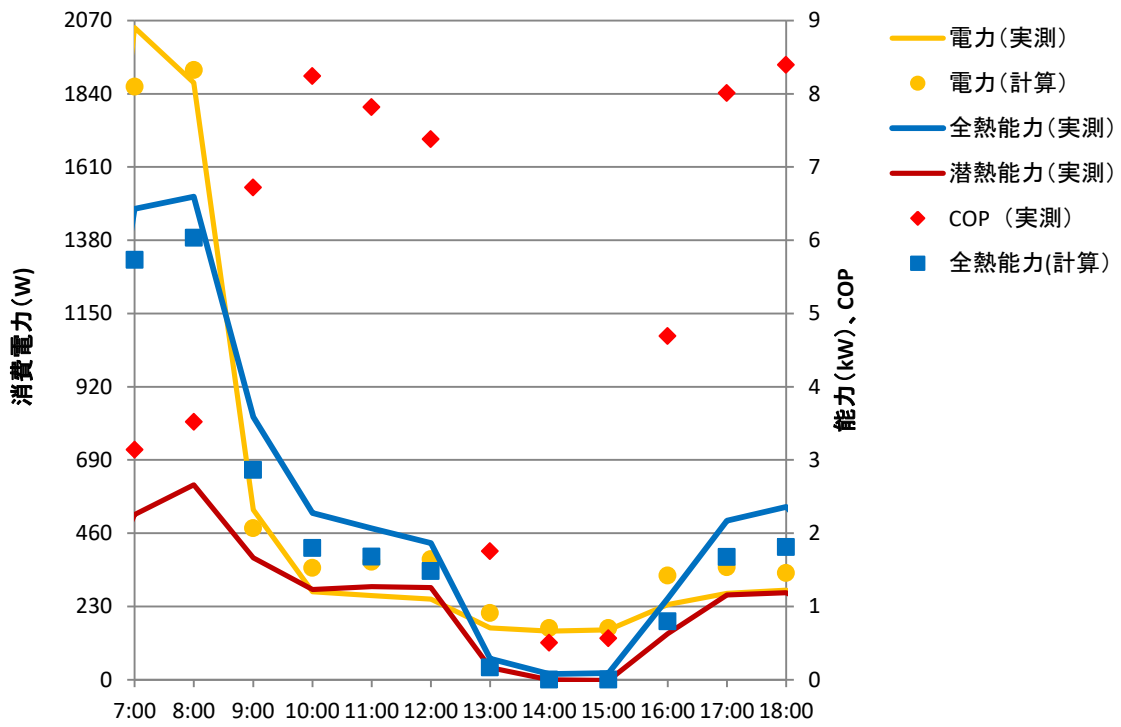


図 239 4/14 運転状況

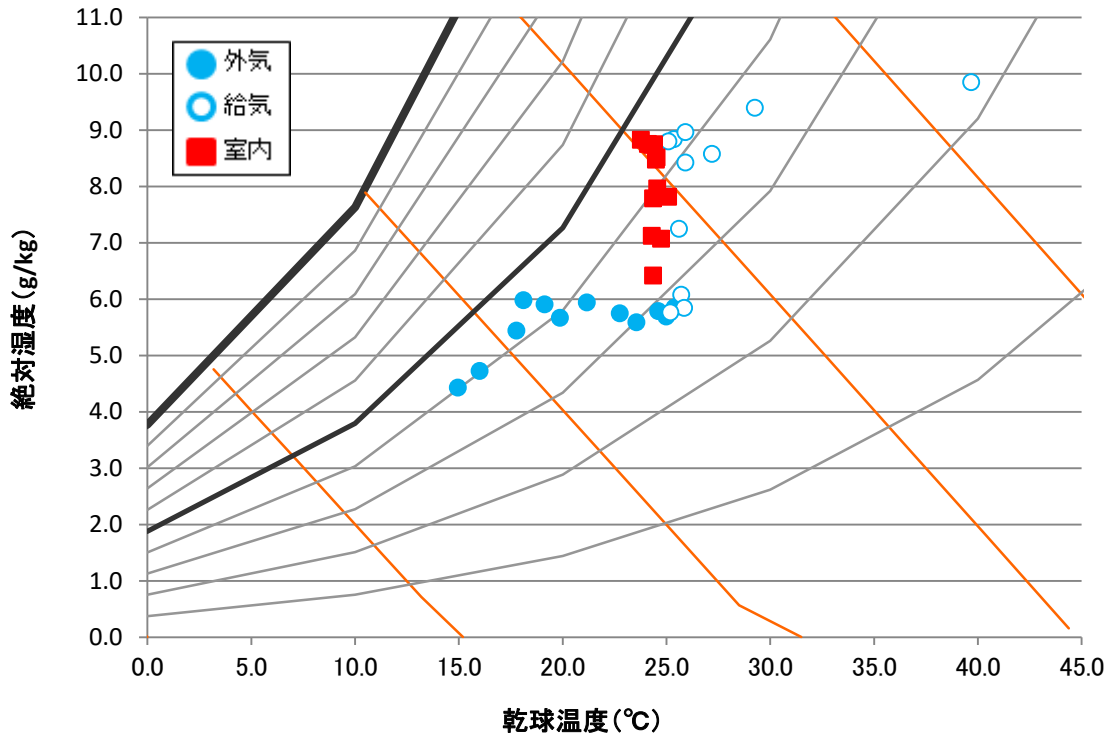


図 240 4/14 空気状況

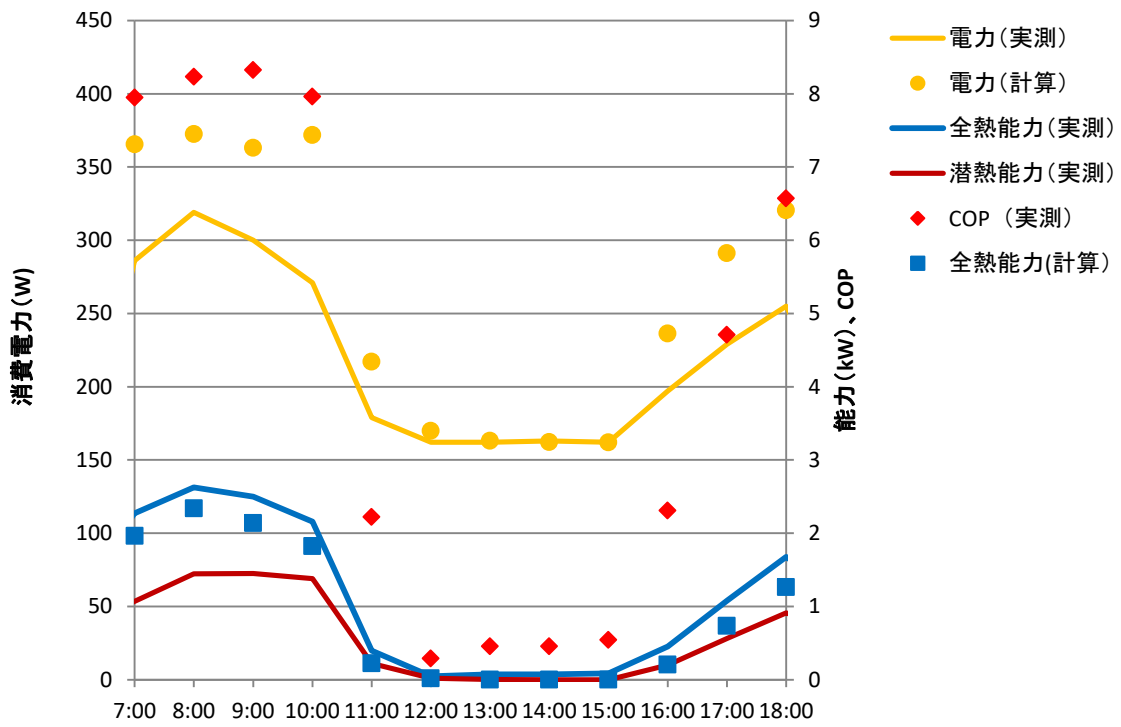


図 241 4/15 運転状況

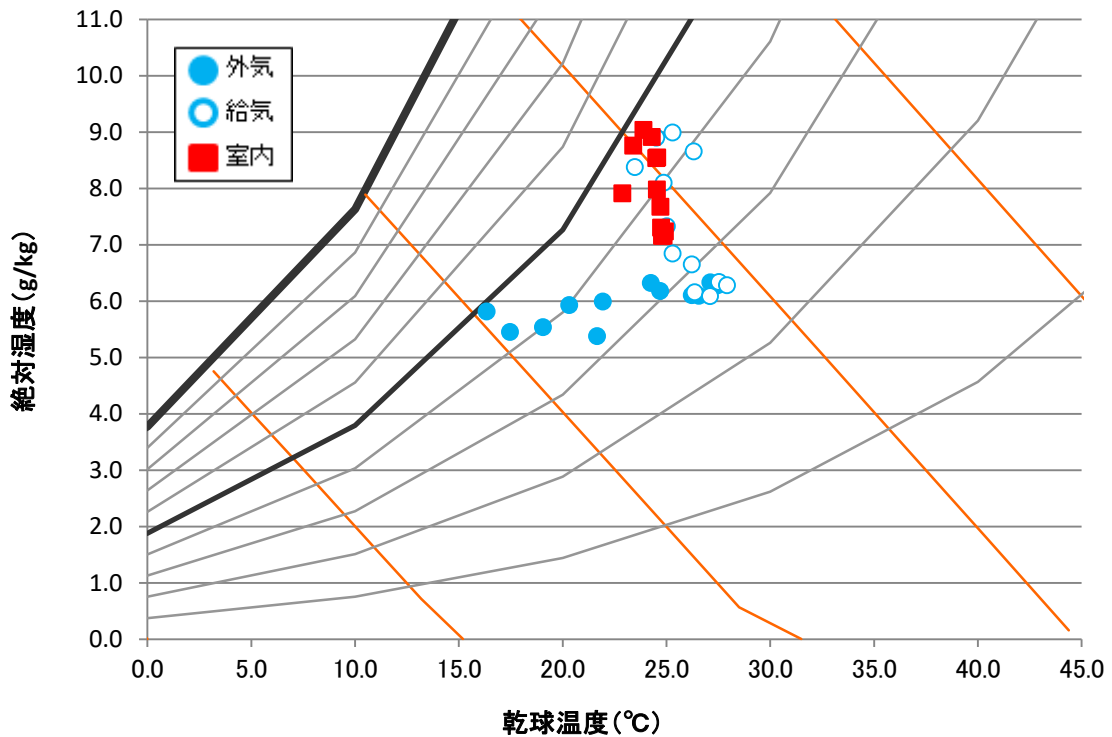


図 242 4/15 空気状況

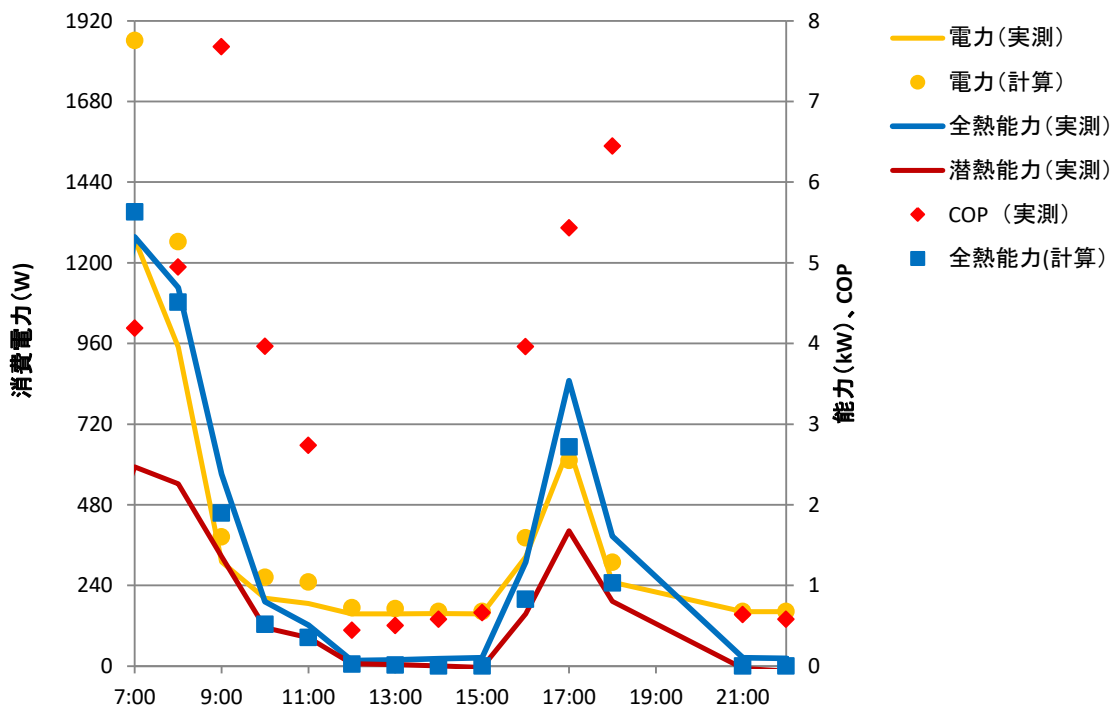


図 243 4/16 運転状況

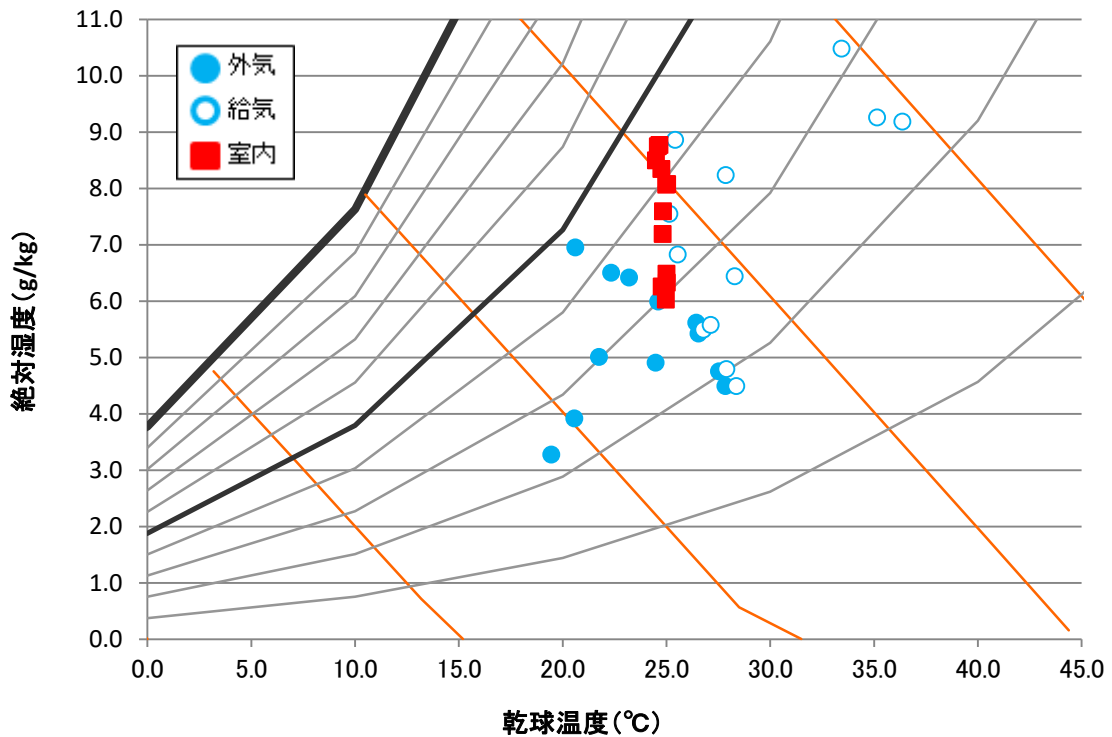


図 244 4/16 空気状況

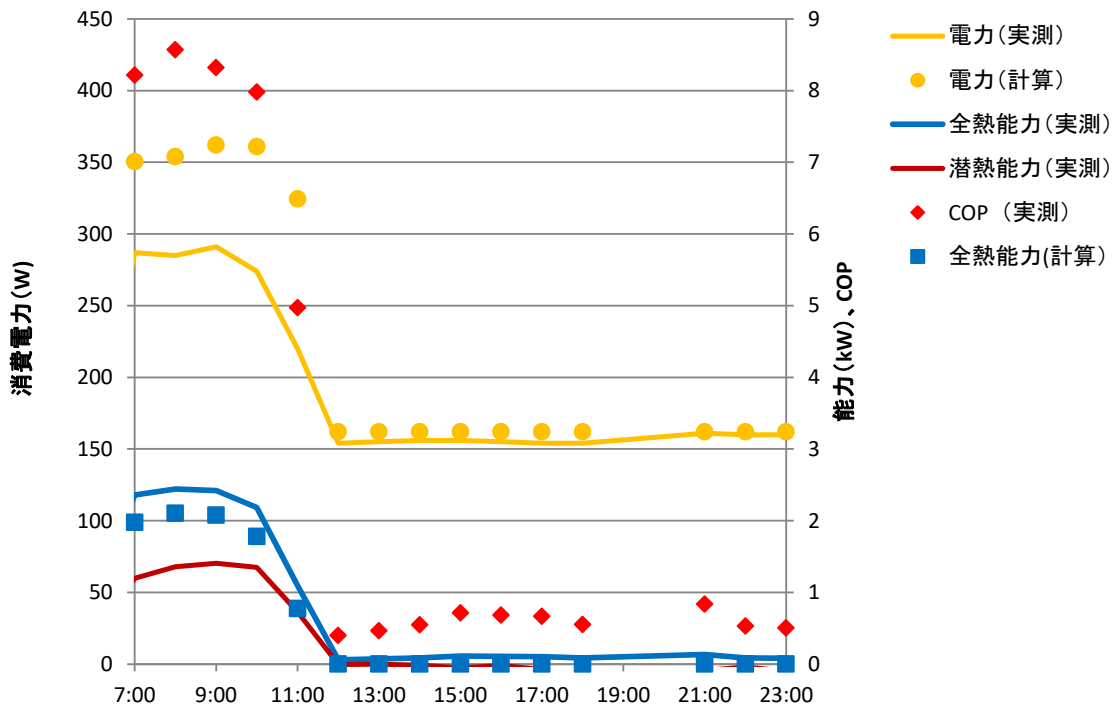


図 245 4/17 運転状況

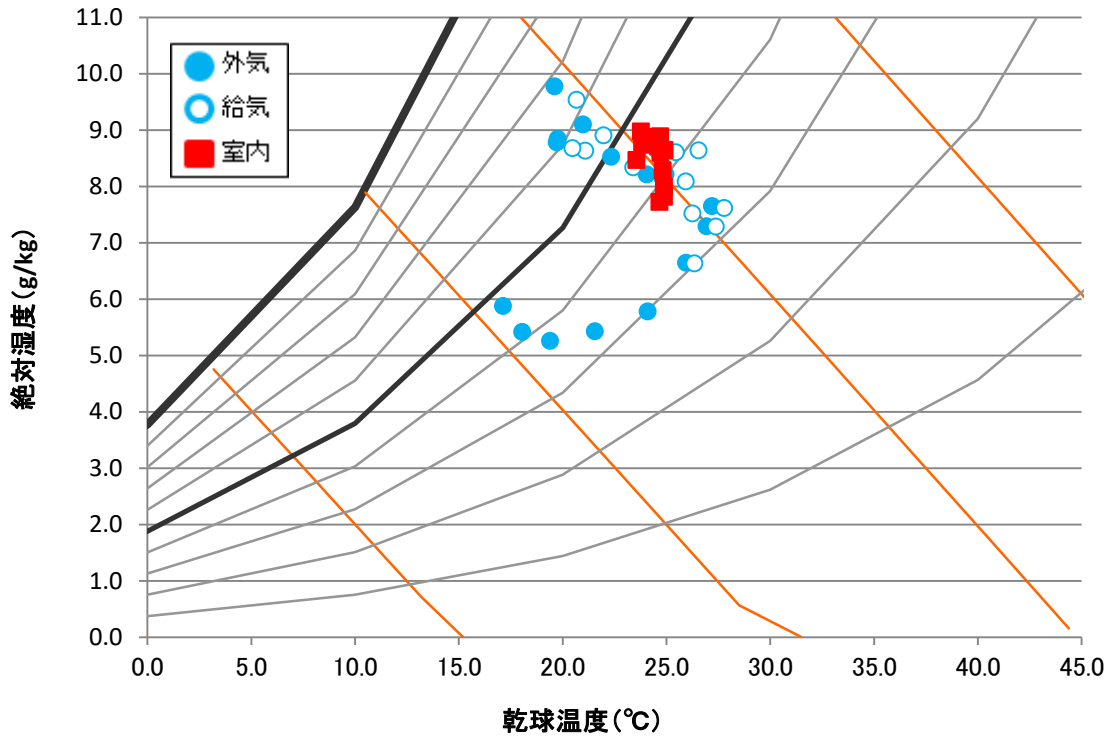


図 246 4/17 空気状況

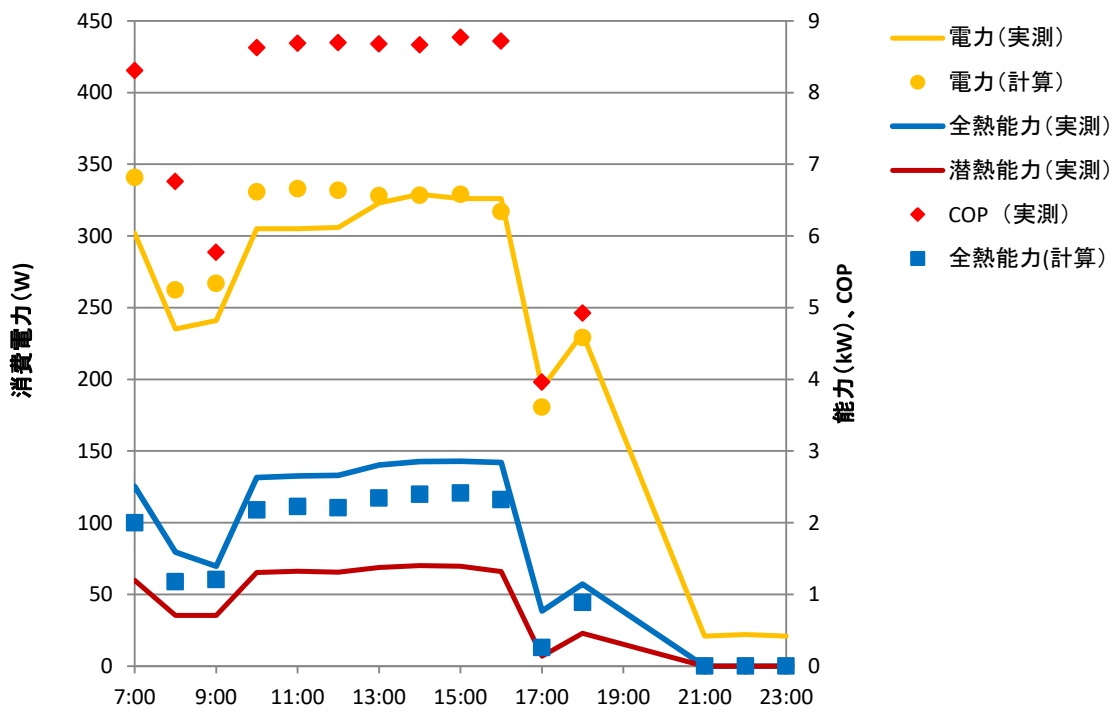


図 247 4/18 運転状況

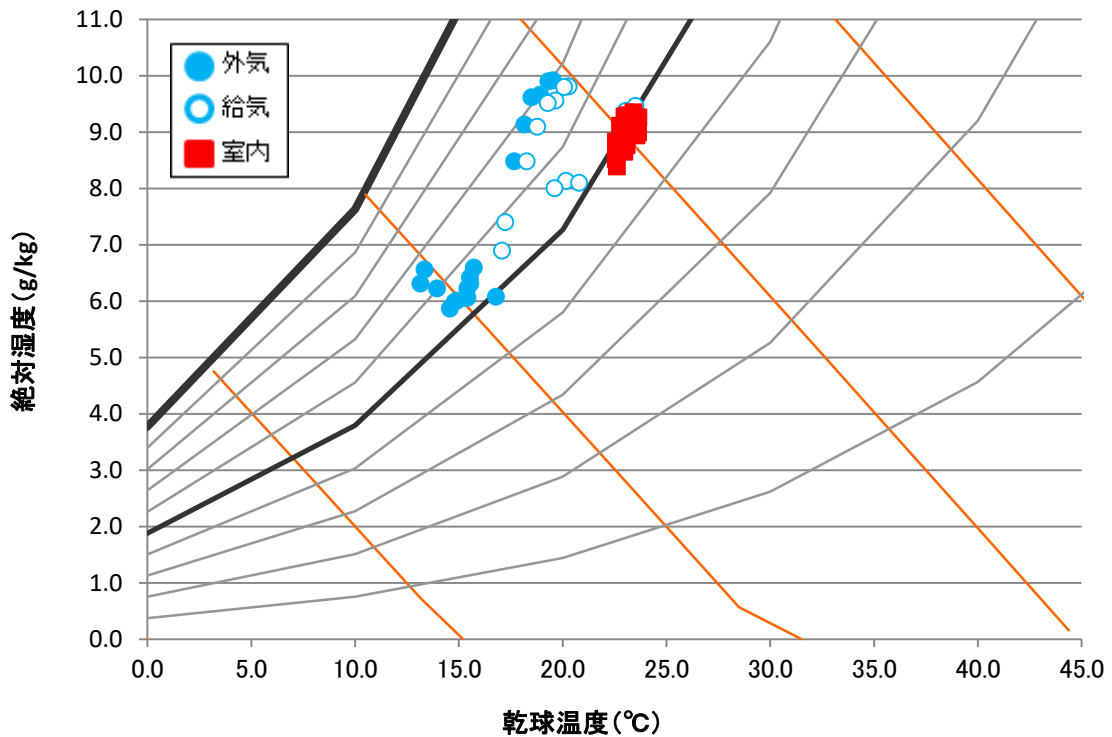


図 248 4/18 空気状況

付録 B

東京 Aビルにおける

ヒートポンプデシカント調湿外調機の圧縮機を入力項とする性能特性式から
計算される消費電力、全熱能力、潜熱能力と実測値の比較データ

データは、日毎の機器の運転状況（消費電力、全熱能力、潜熱能力）の実測値と計算値の1時間毎の平均値を示す。

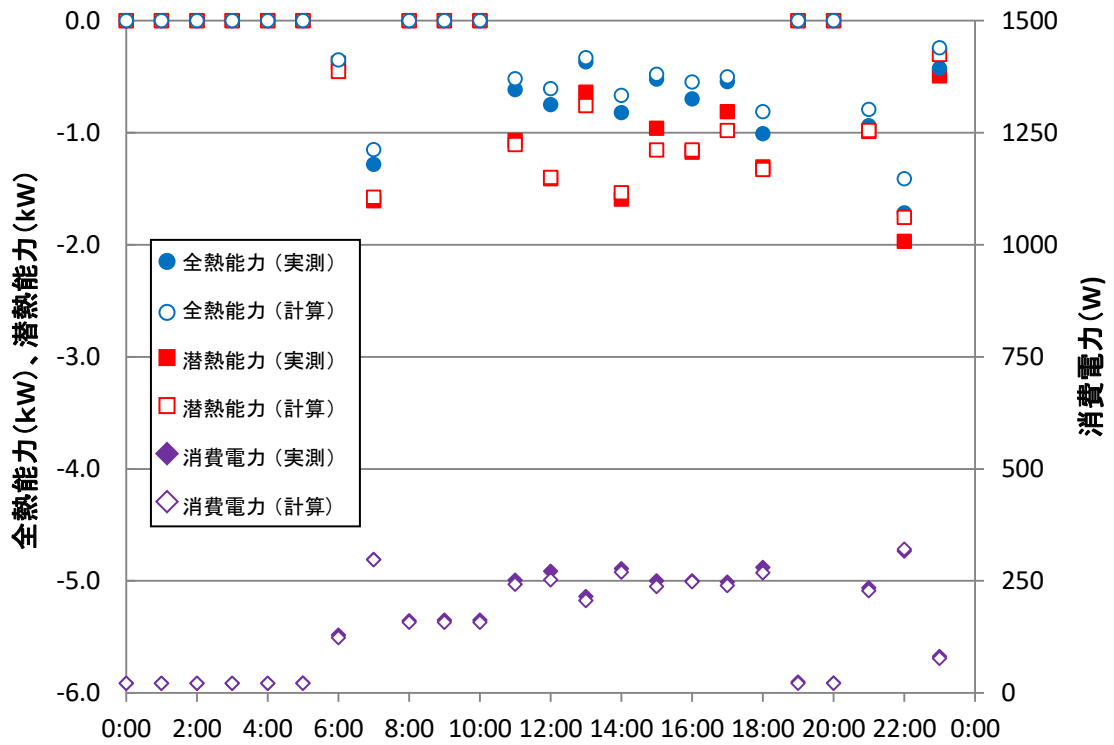


図 1 実測結果, 計算結果比較 (6/2)

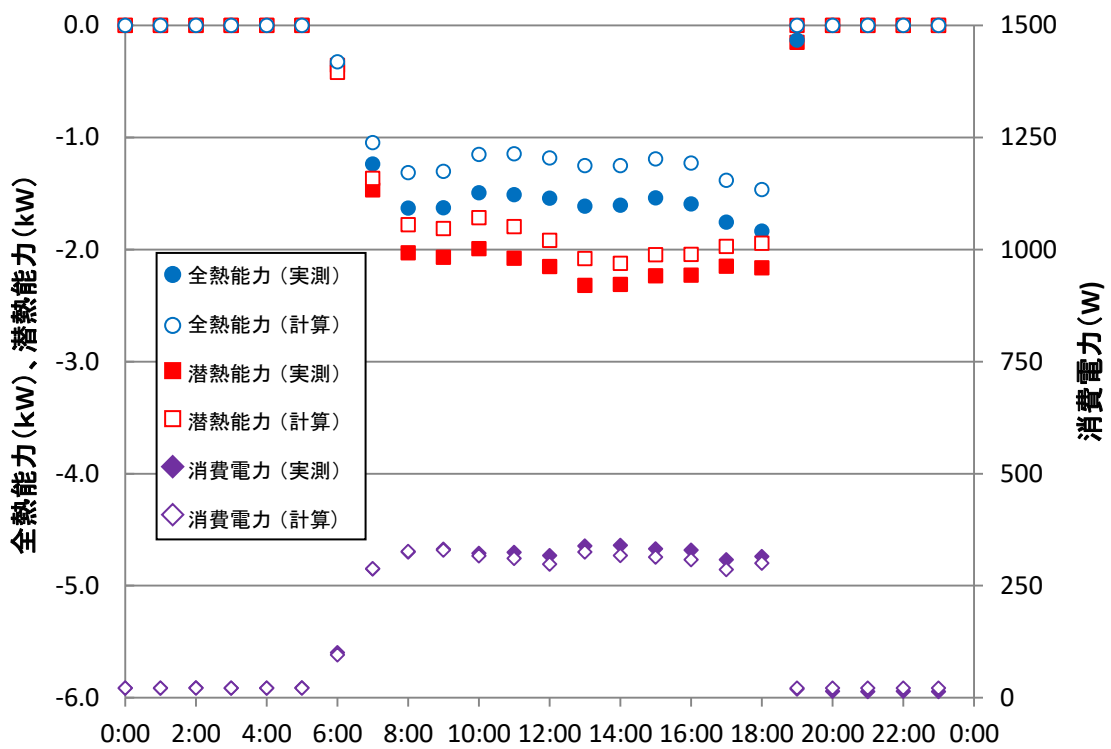


図 2 実測結果, 計算結果比較 (6/3)

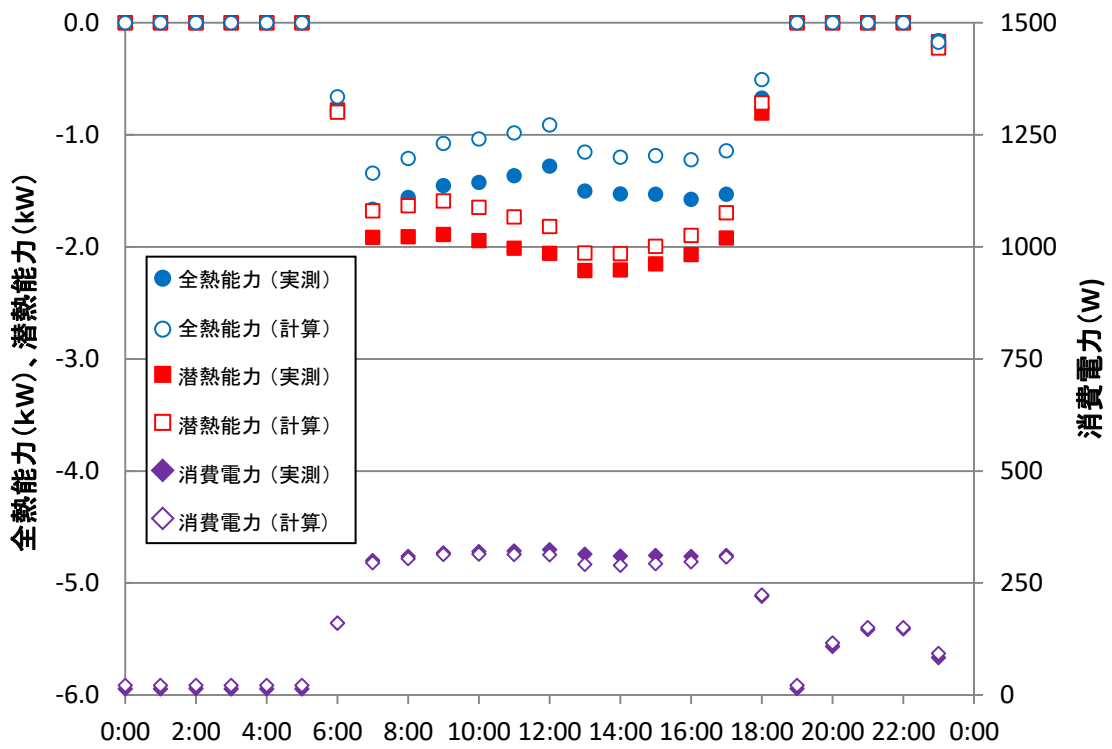


圖 3 實測結果，計算結果比較 (6/4)

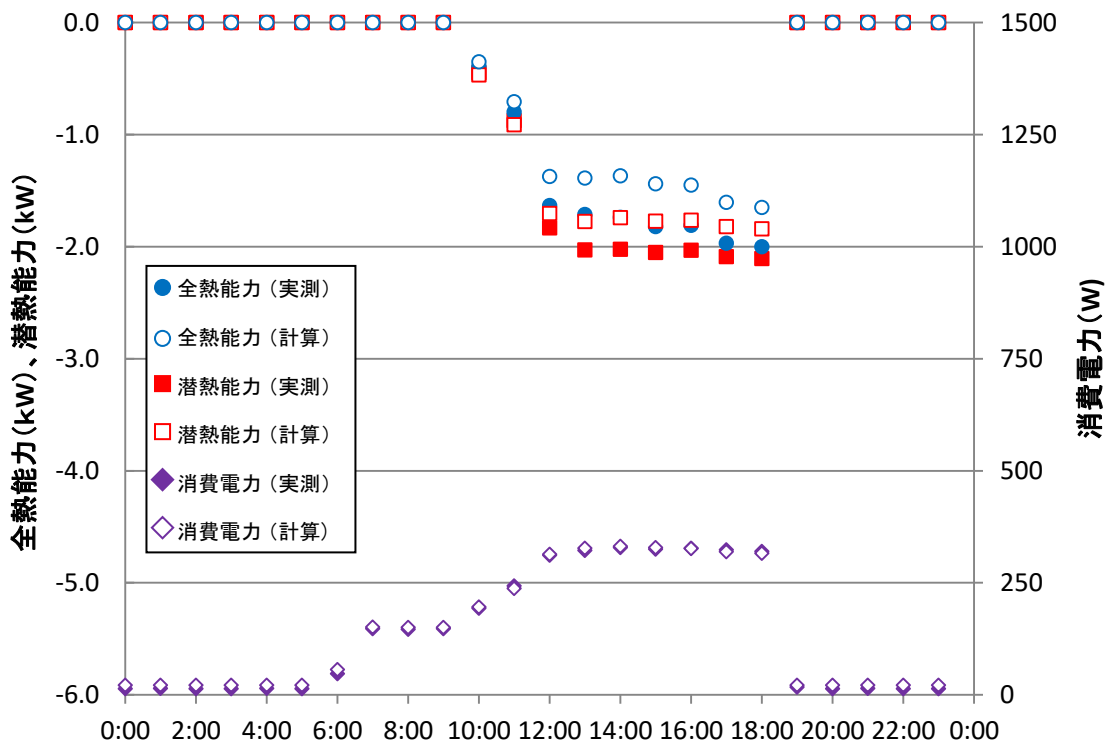


圖 4 實測結果，計算結果比較 (6/5)

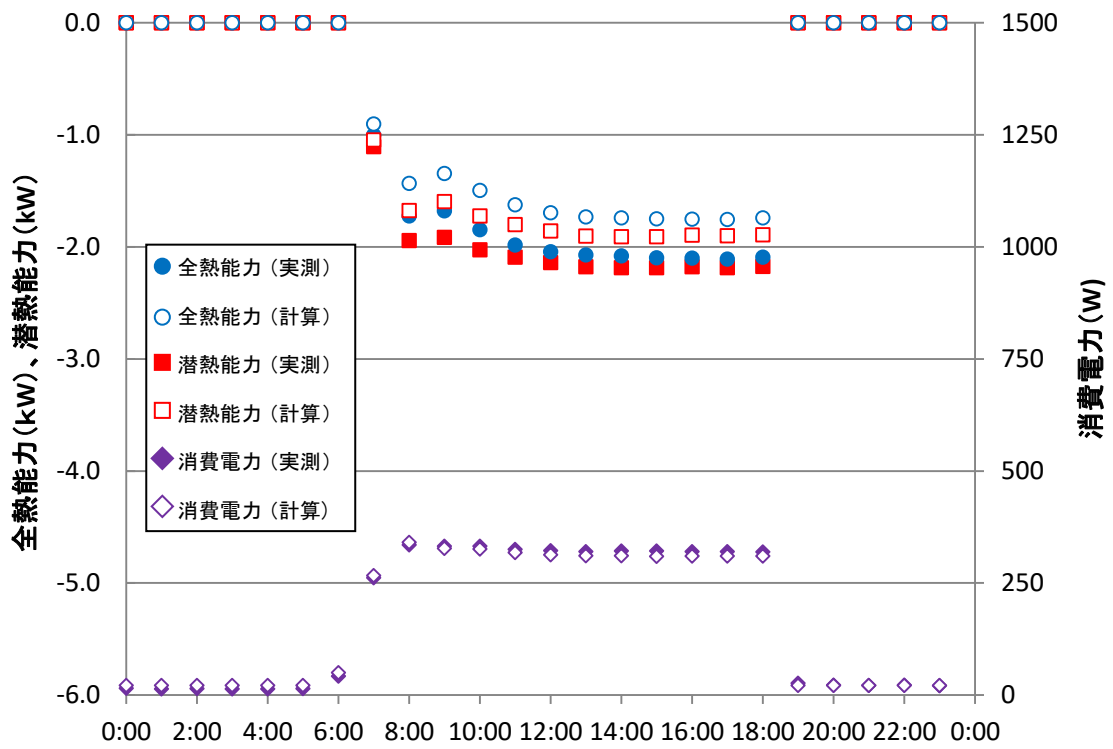


図 5 実測結果，計算結果比較 (6/6)

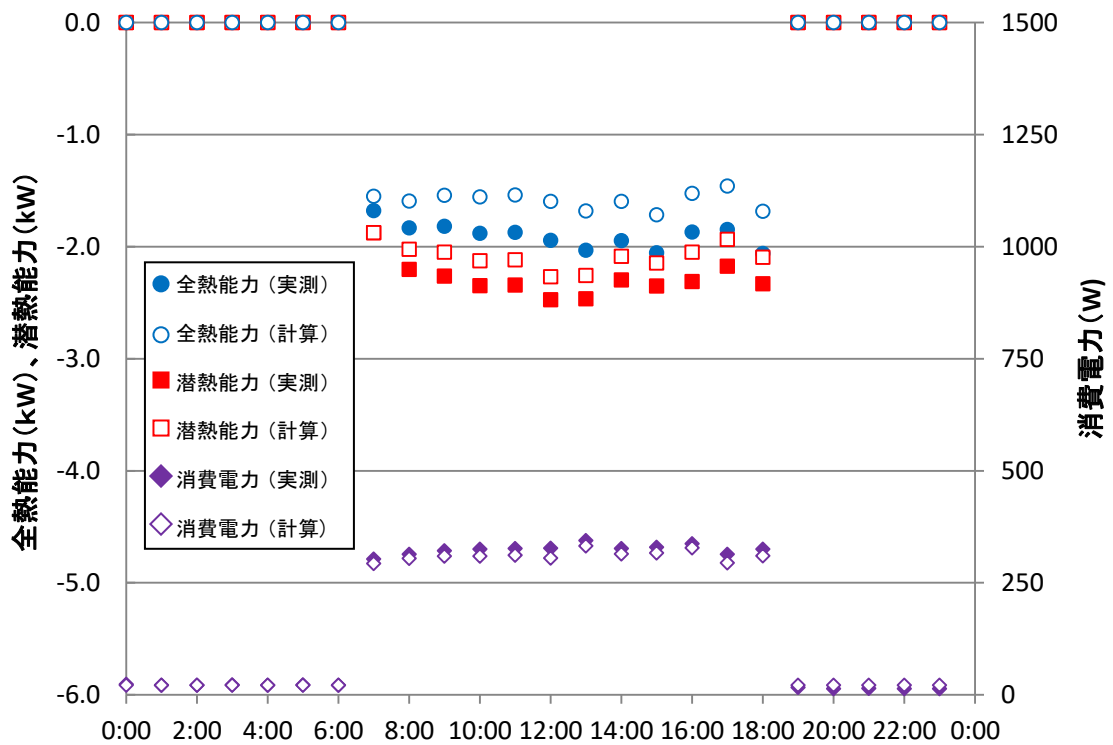


図 6 実測結果，計算結果比較 (6/9)

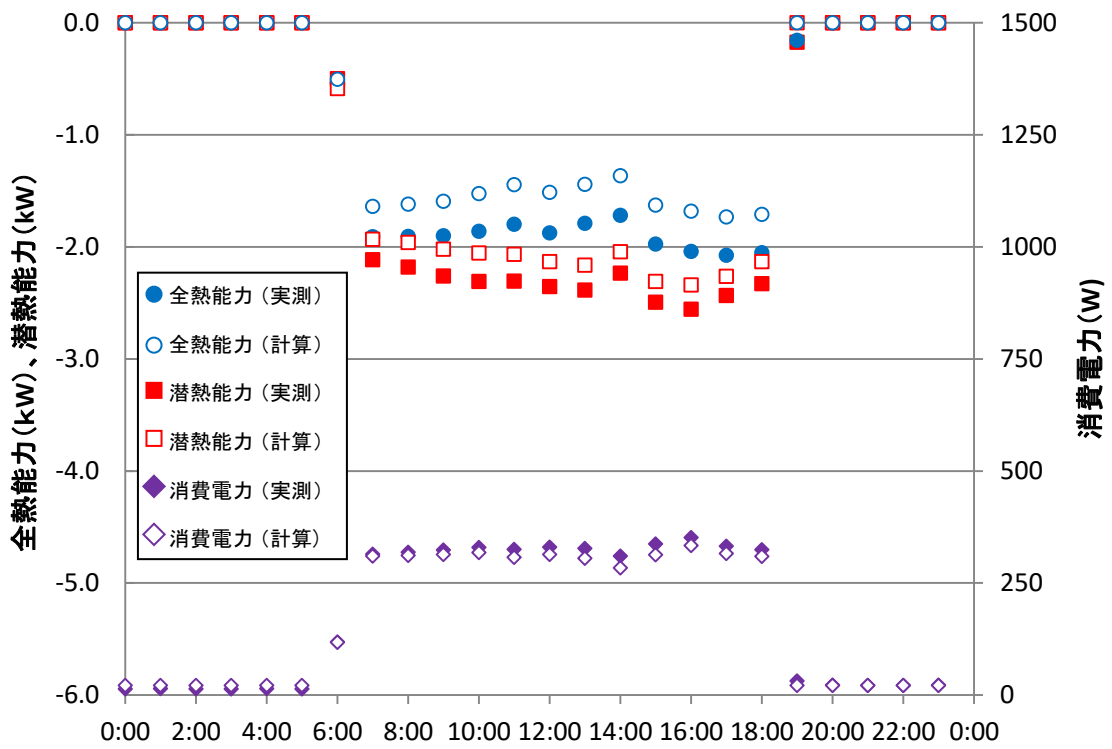


図 7 実測結果, 計算結果比較 (6/10)

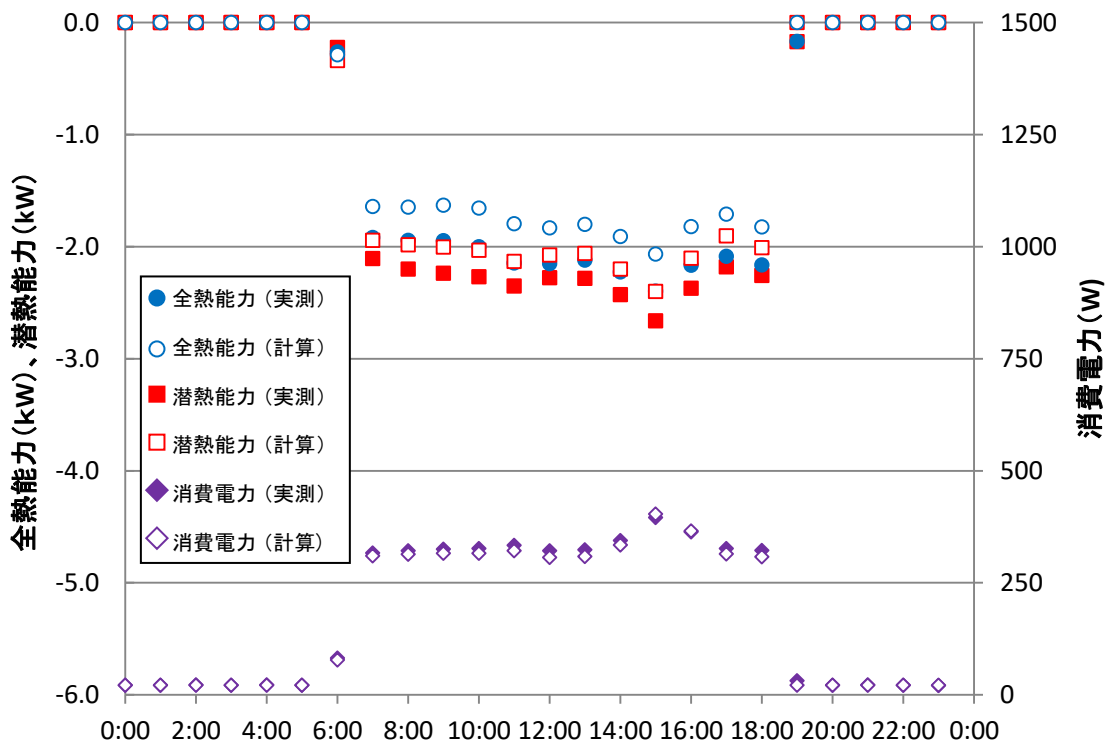


図 8 実測結果, 計算結果比較 (6/11)

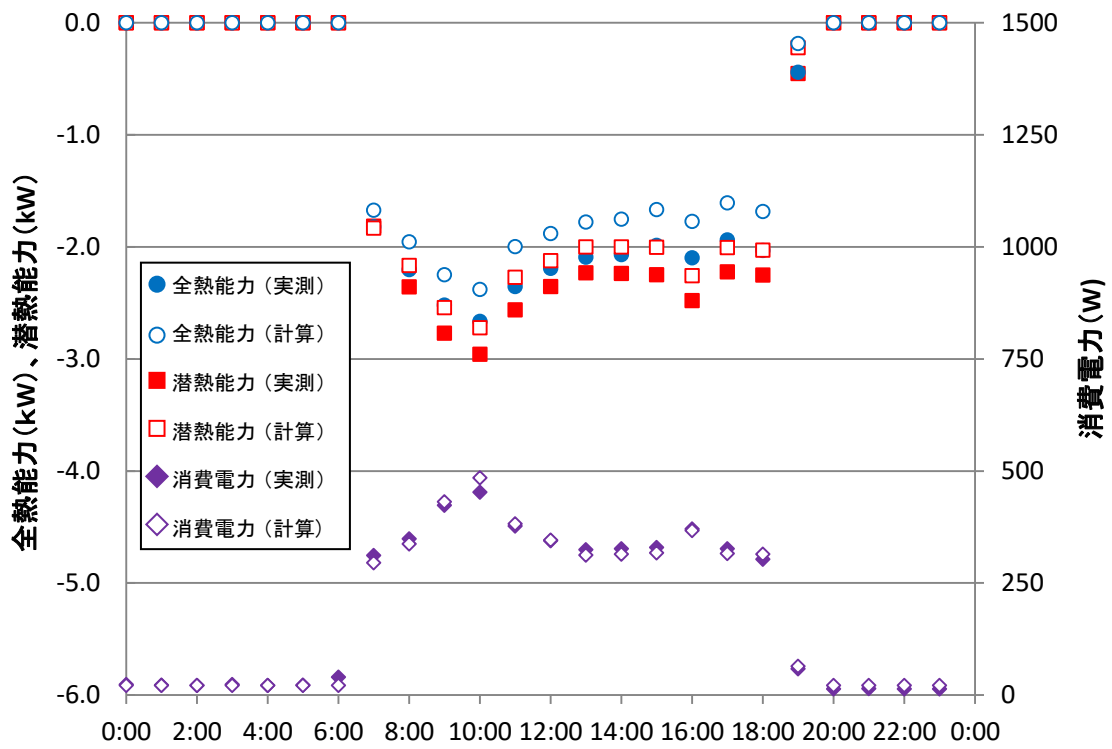


图 9 実測結果，計算結果比較 (6/12)

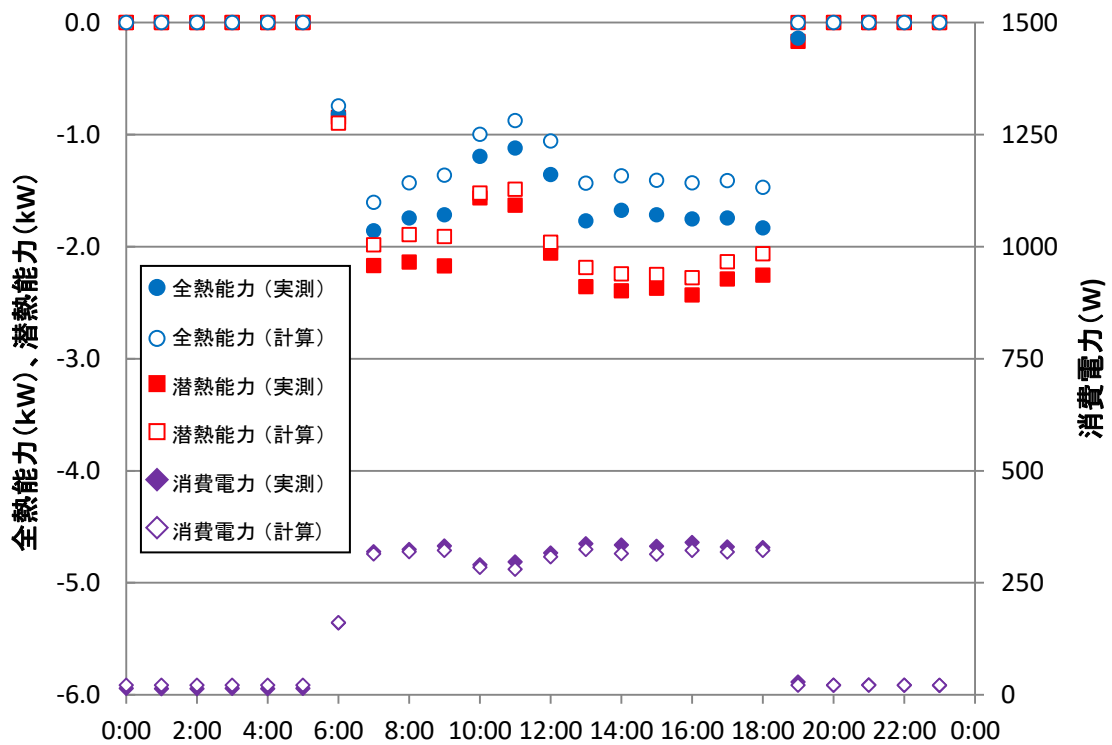


图 10 実測結果，計算結果比較 (6/13)

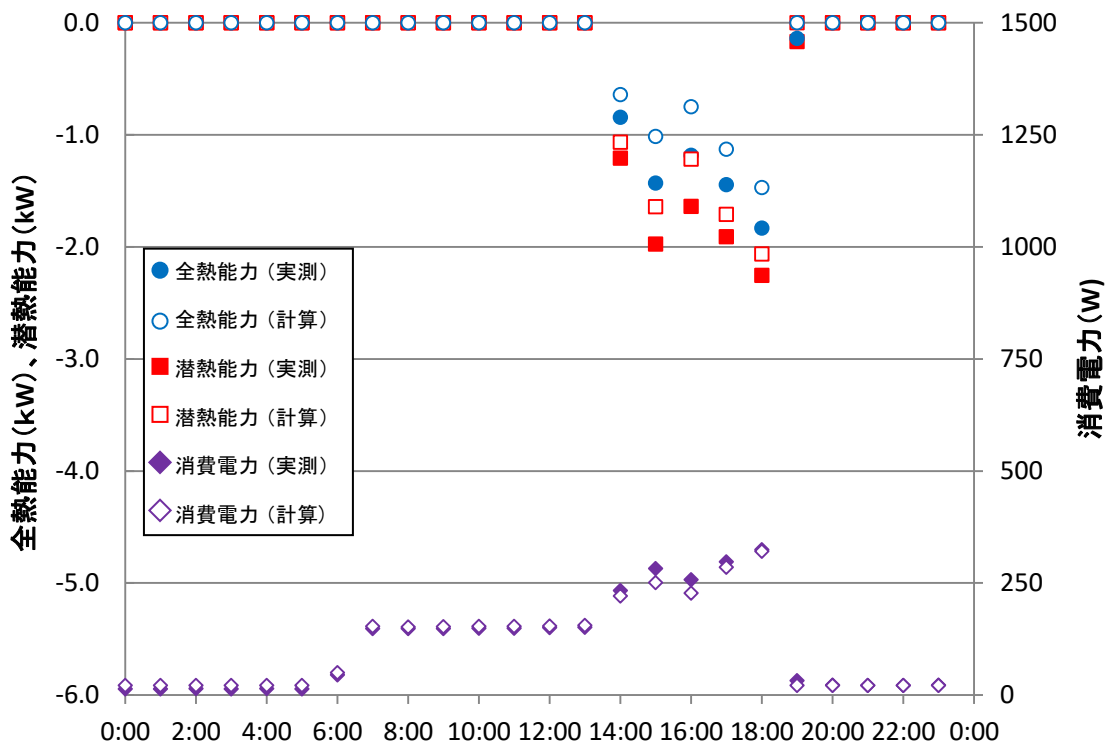


図 11 実測結果，計算結果比較 (6/16)

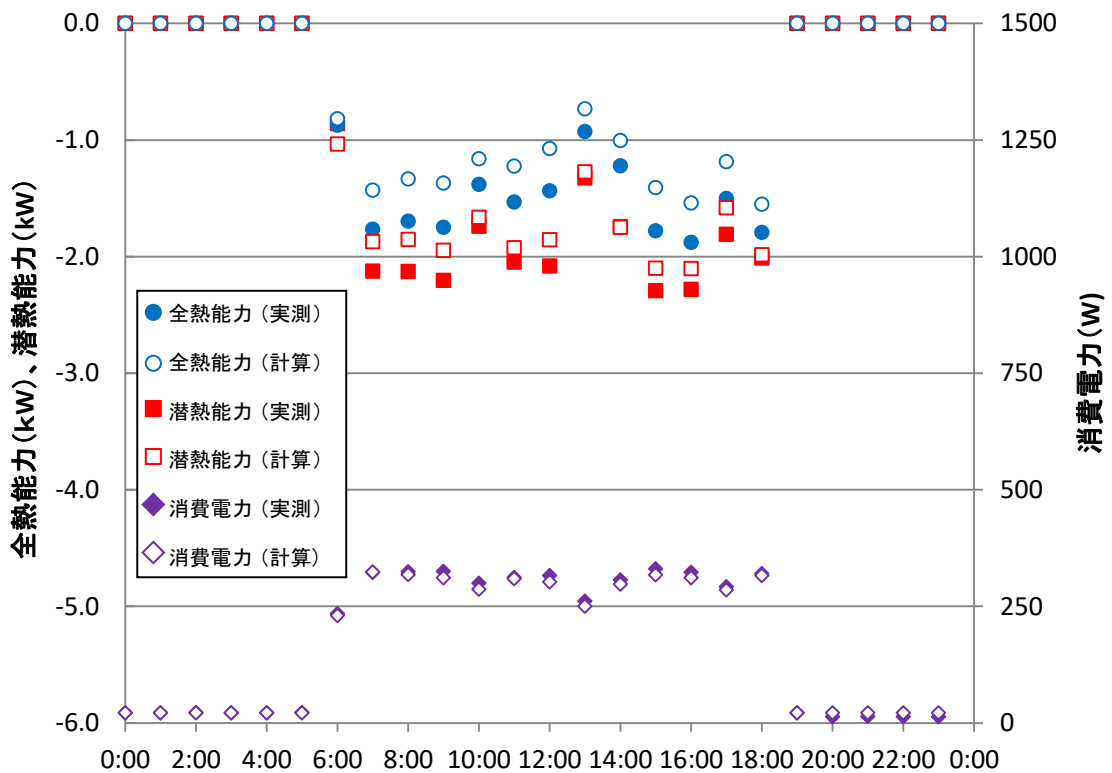


図 12 実測結果，計算結果比較 (6/17)

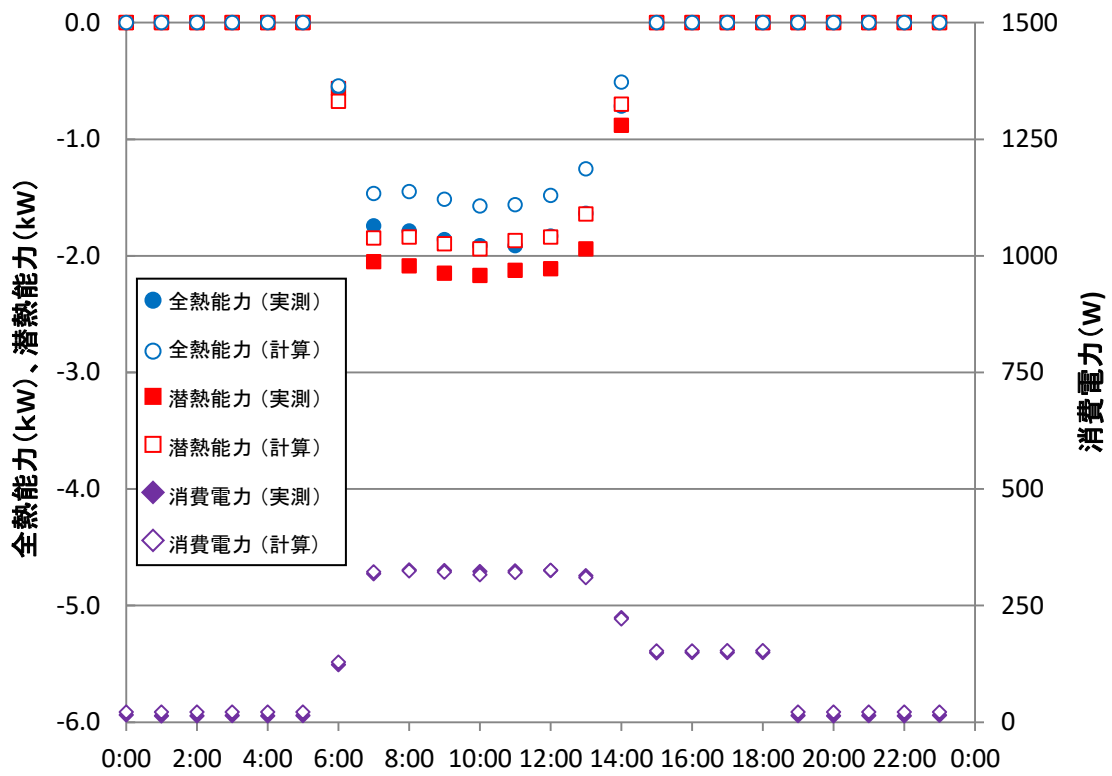


圖 13 實測結果，計算結果比較 (6/18)

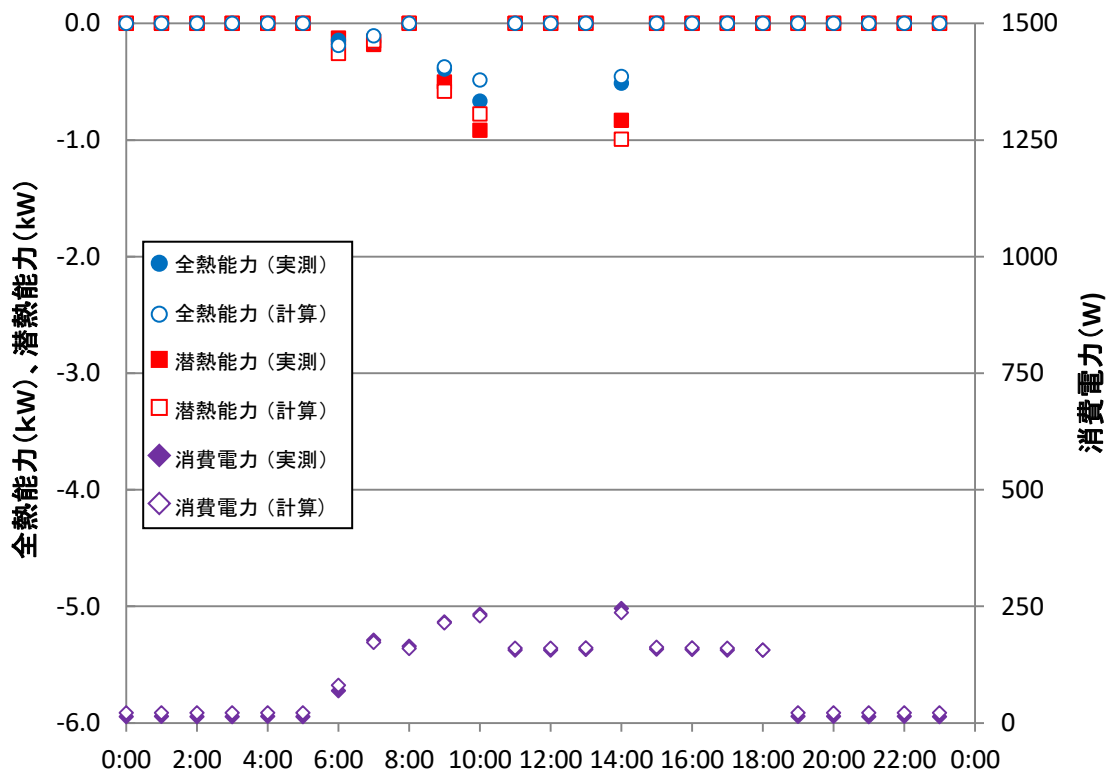


圖 14 實測結果，計算結果比較 (6/19)

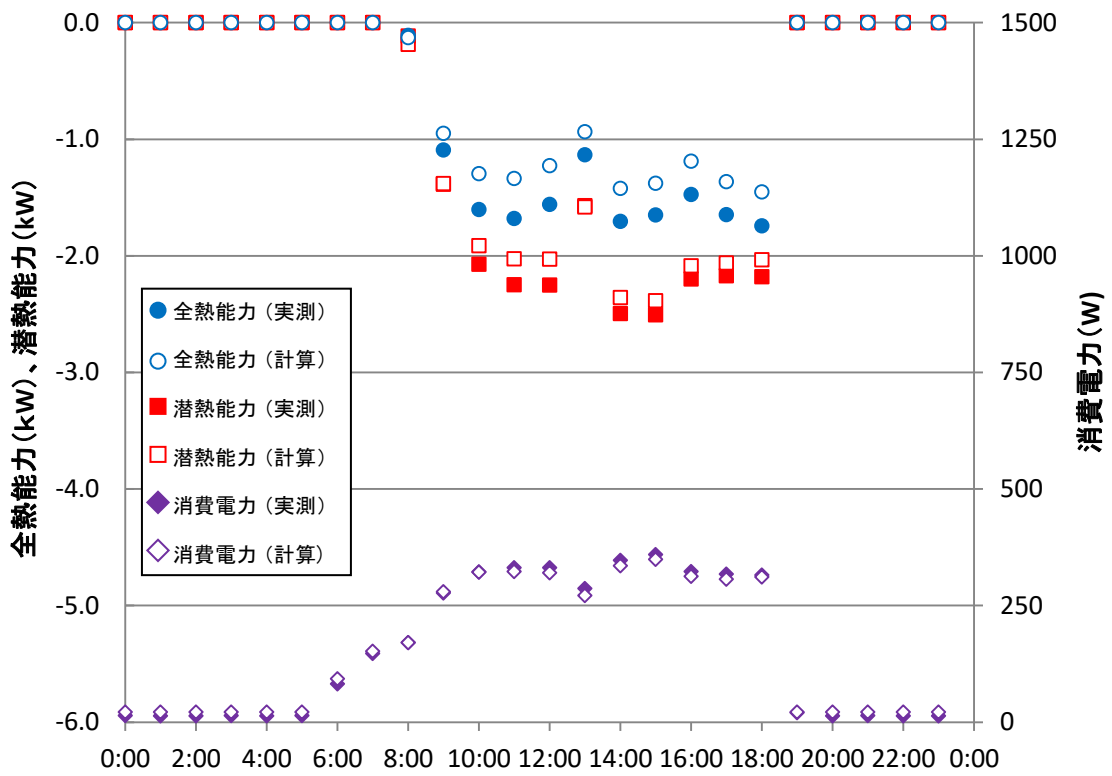


図 15 実測結果，計算結果比較 (6/20)

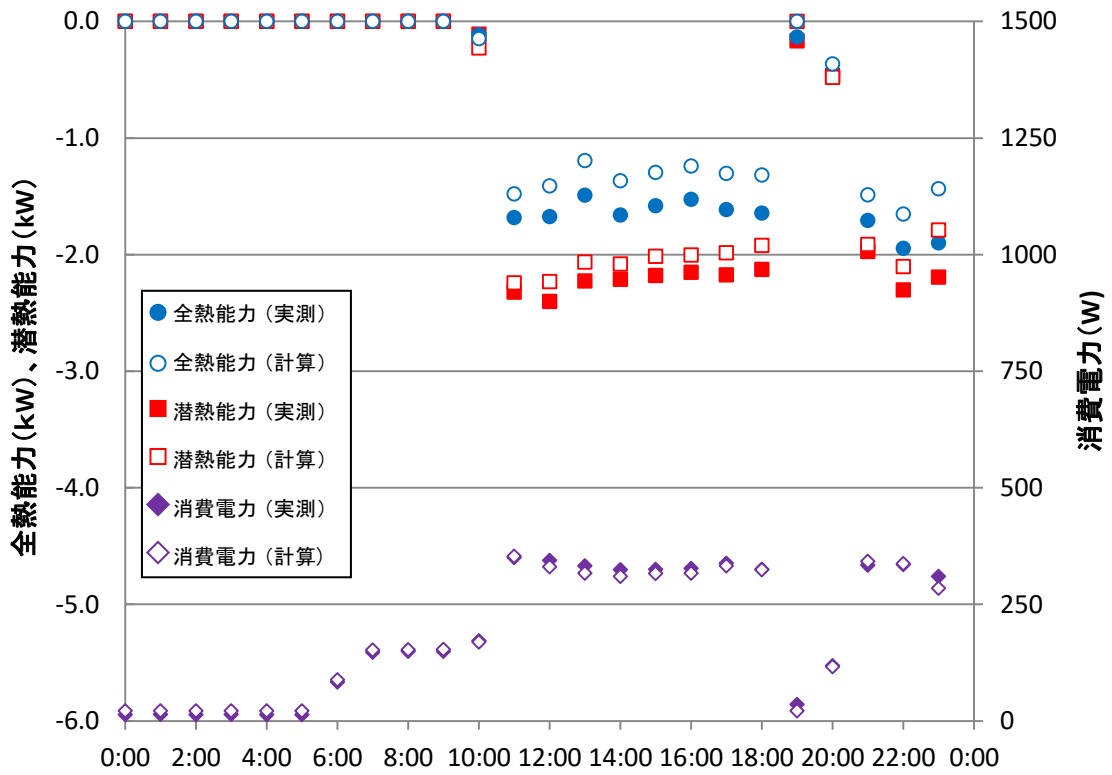


図 16 実測結果，計算結果比較 (6/23)

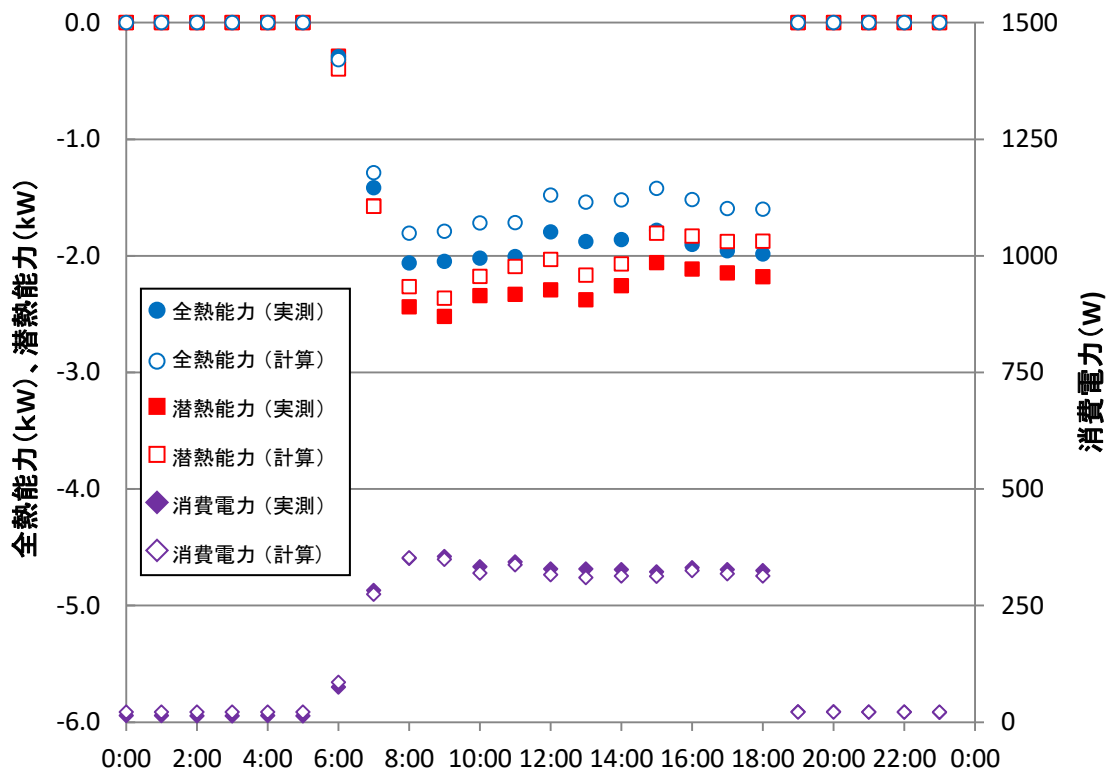


図 17 実測結果，計算結果比較 (6/24)

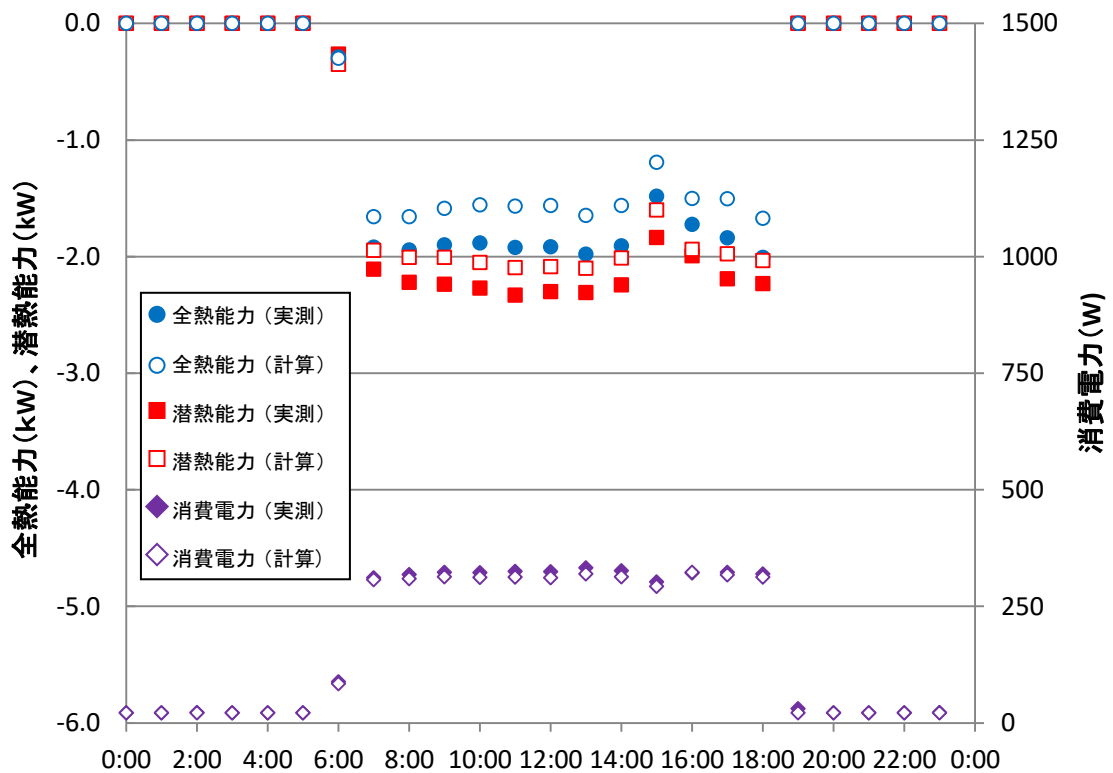


図 18 実測結果，計算結果比較 (6/25)

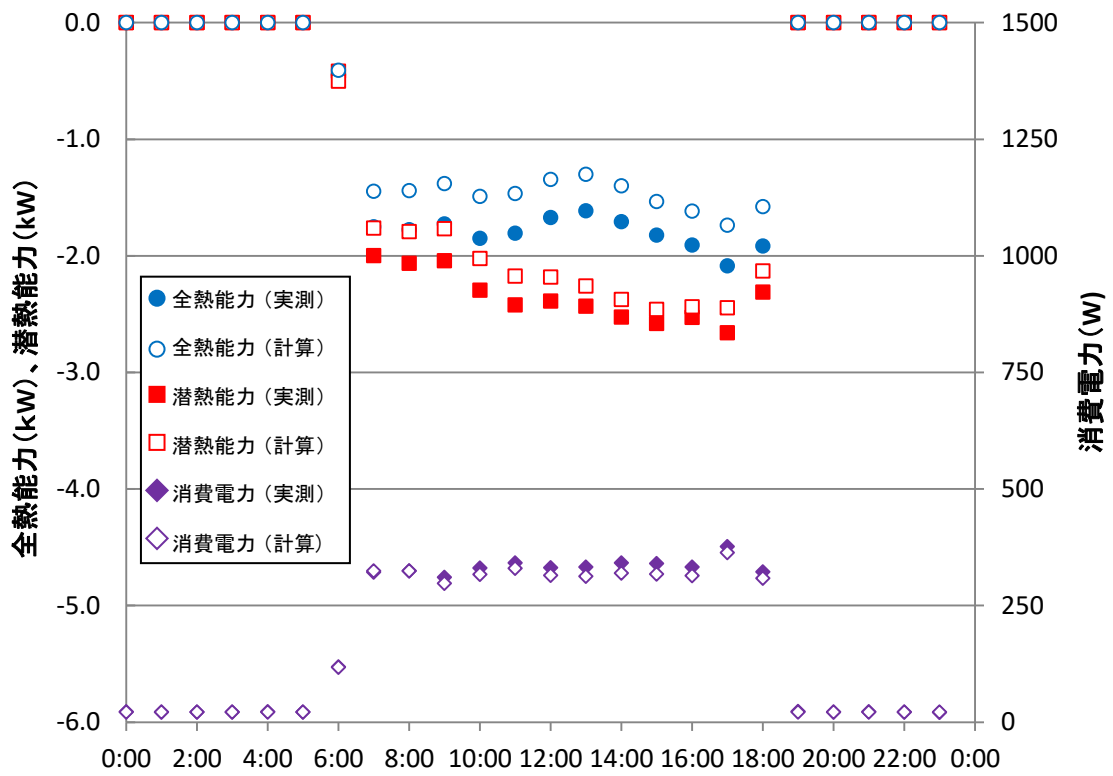


圖 19 實測結果，計算結果比較 (6/26)

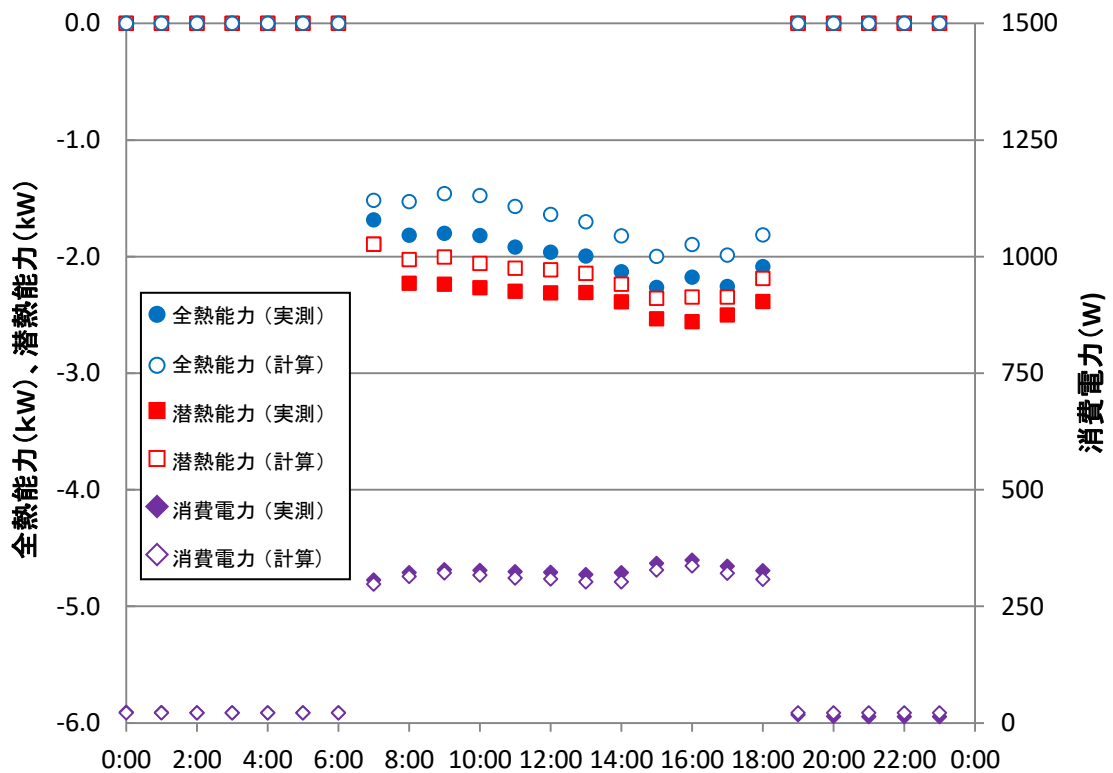


圖 20 實測結果，計算結果比較 (6/27)

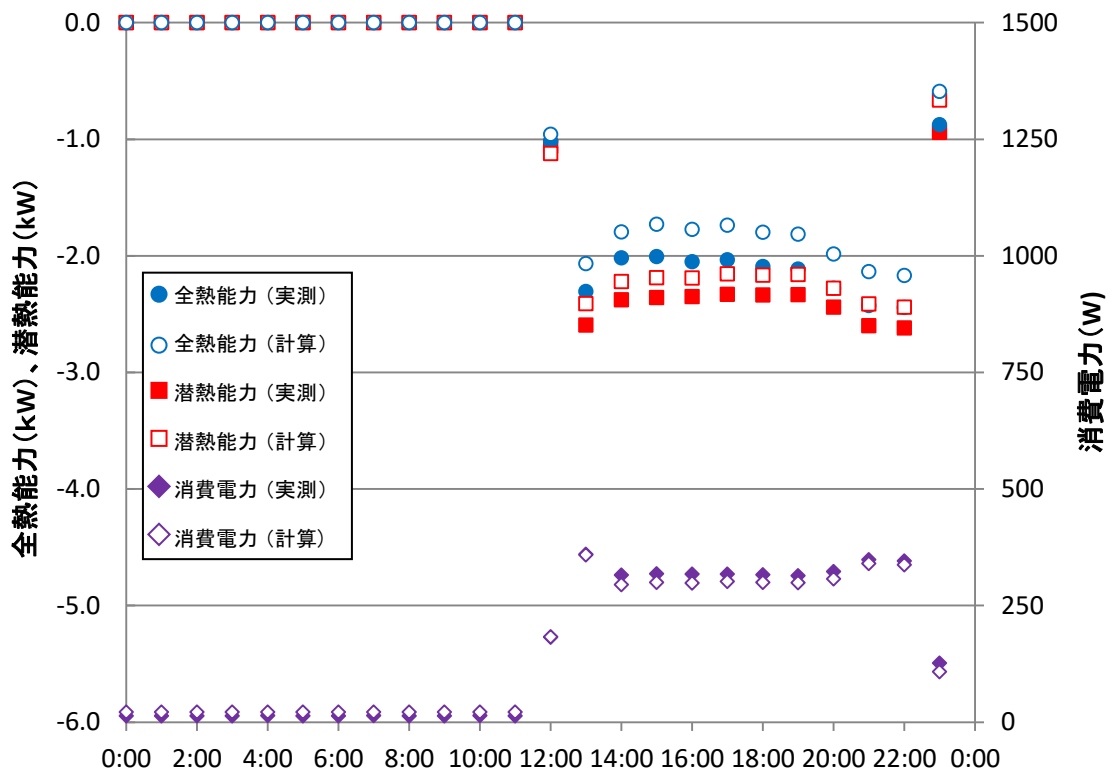


圖 21 實測結果，計算結果比較 (6/28)

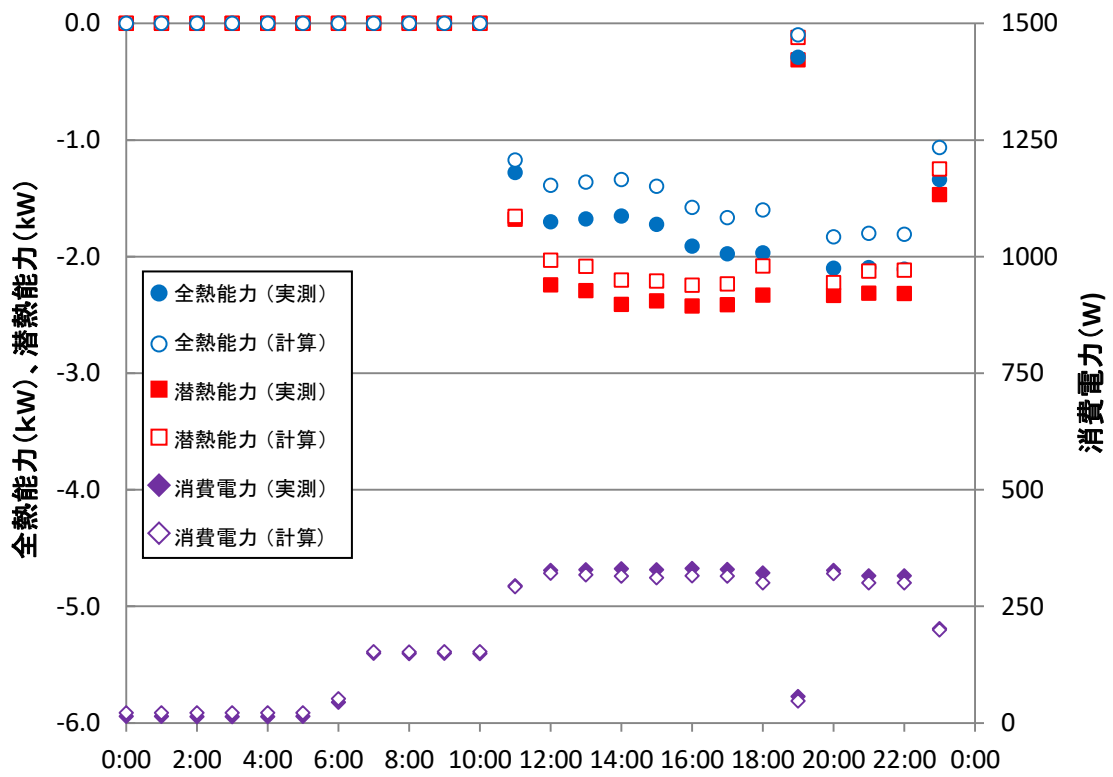


圖 22 實測結果，計算結果比較 (6/30)

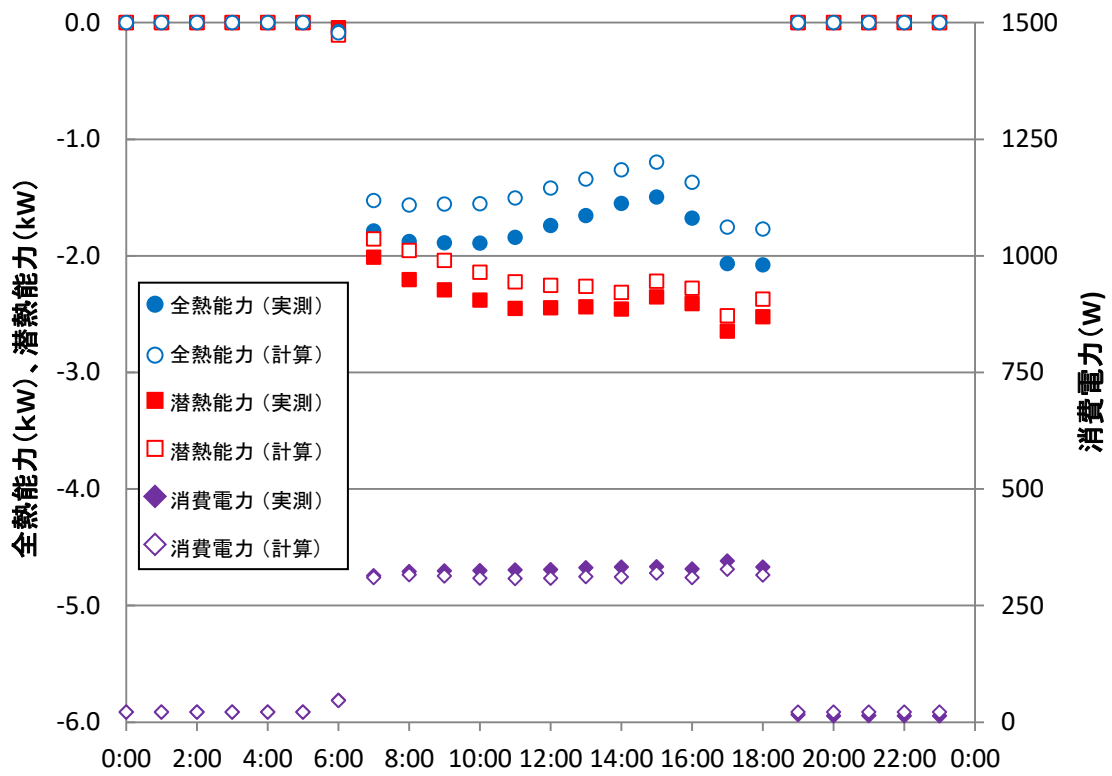


図 23 実測結果, 計算結果比較 (7/1)

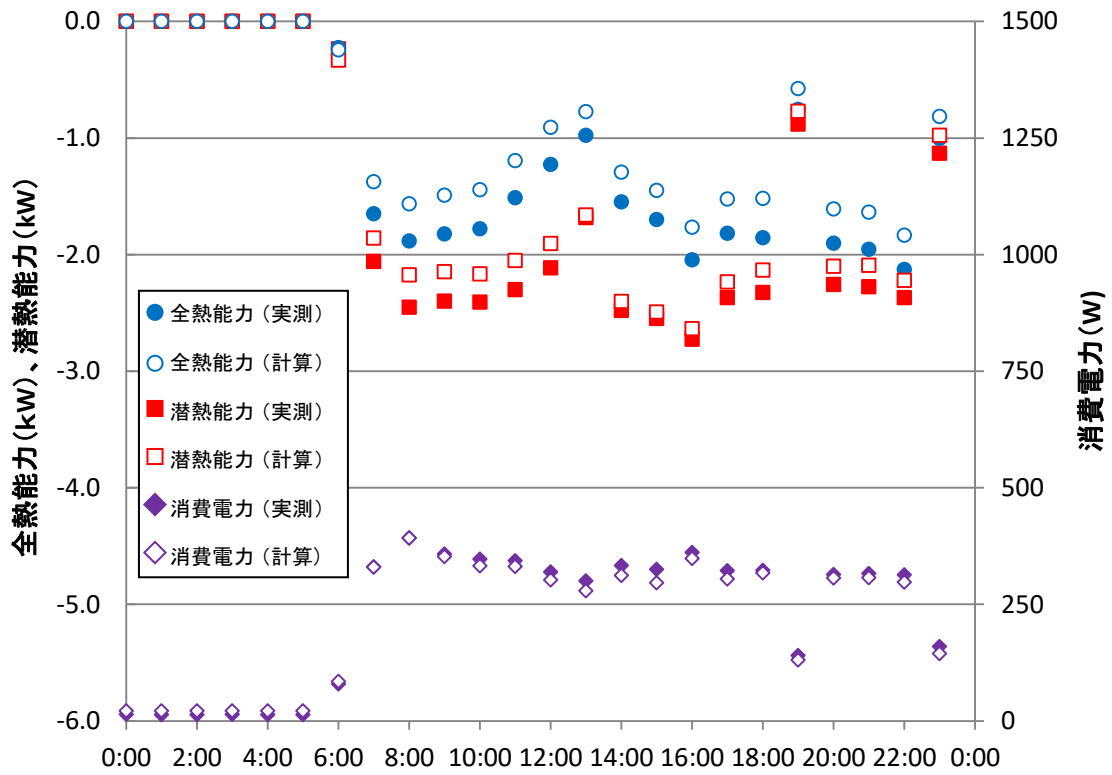


図 24 実測結果, 計算結果比較 (7/2)

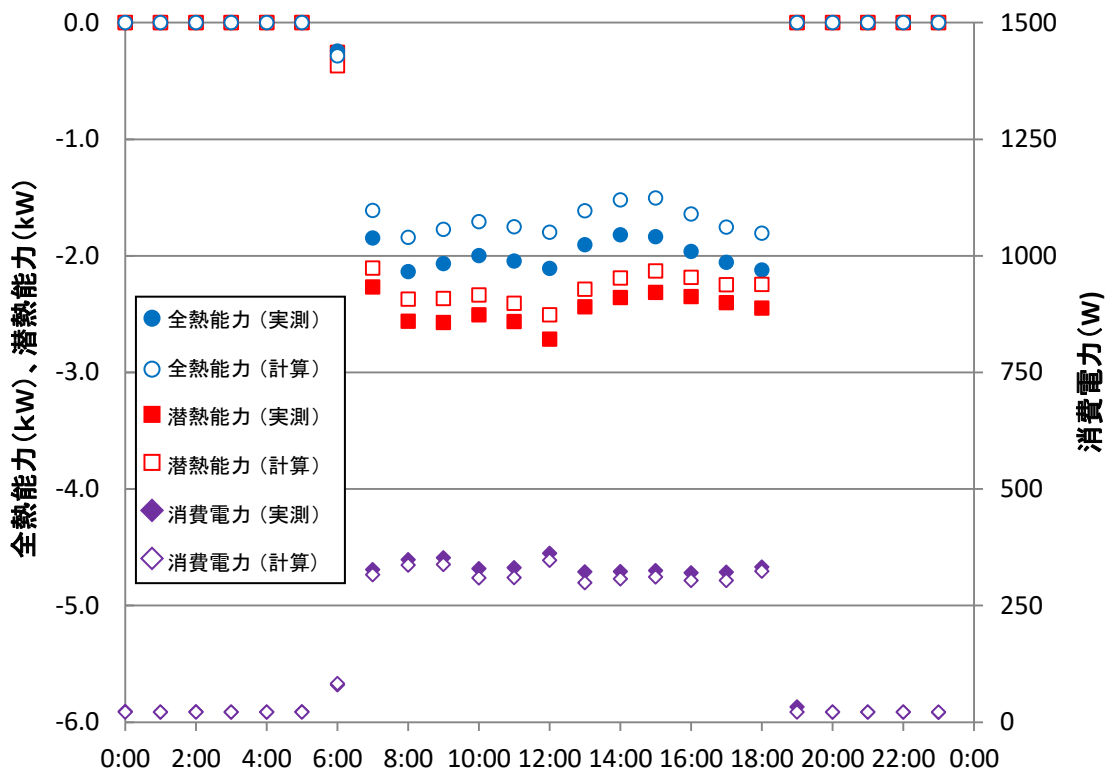


図 25 実測結果，計算結果比較 (7/3)

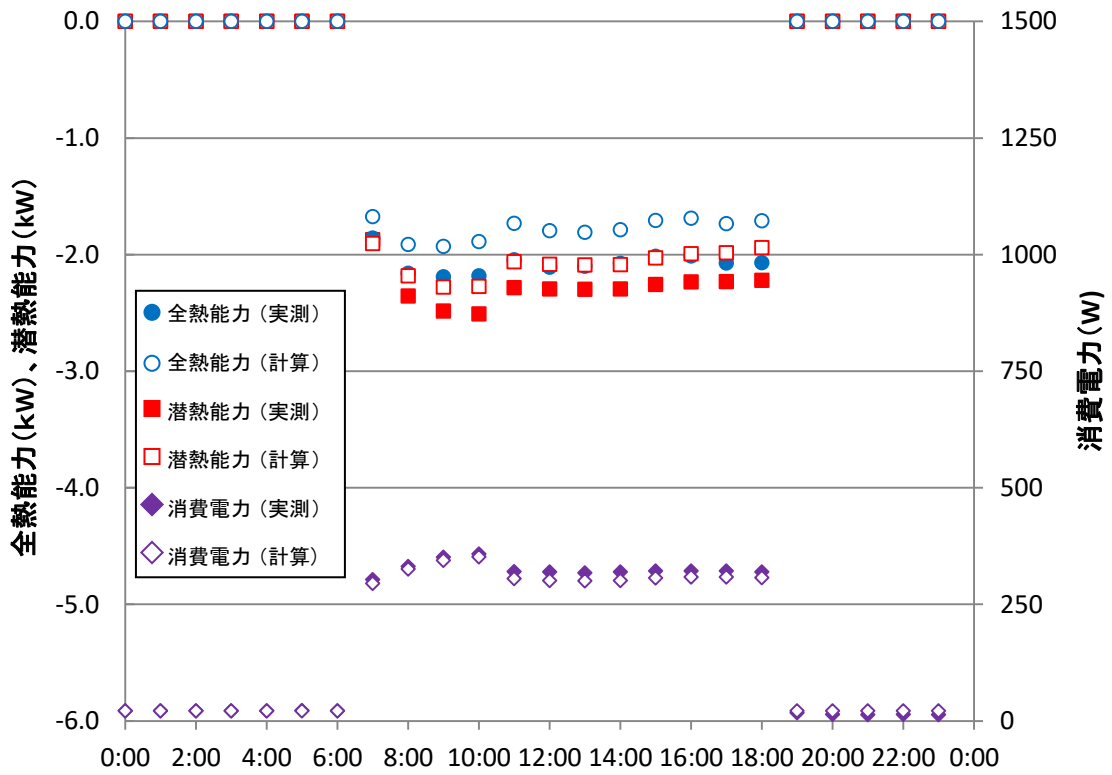


図 26 実測結果，計算結果比較 (7/4)

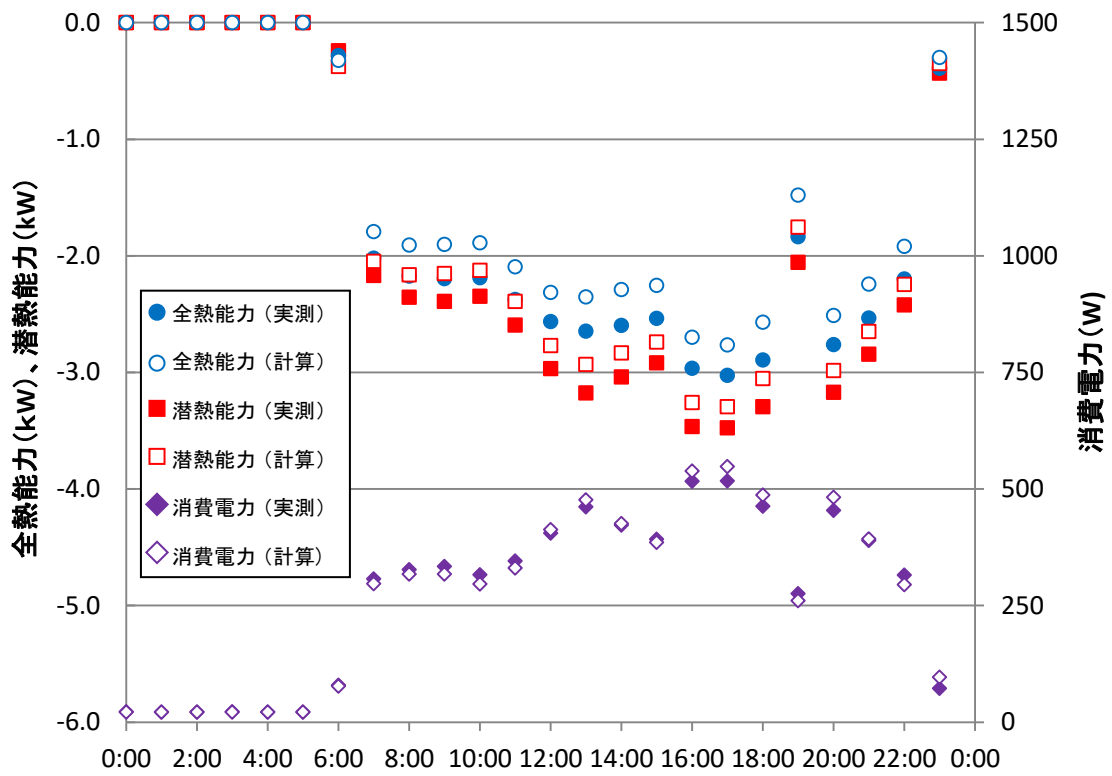


図 27 実測結果，計算結果比較 (7/7)

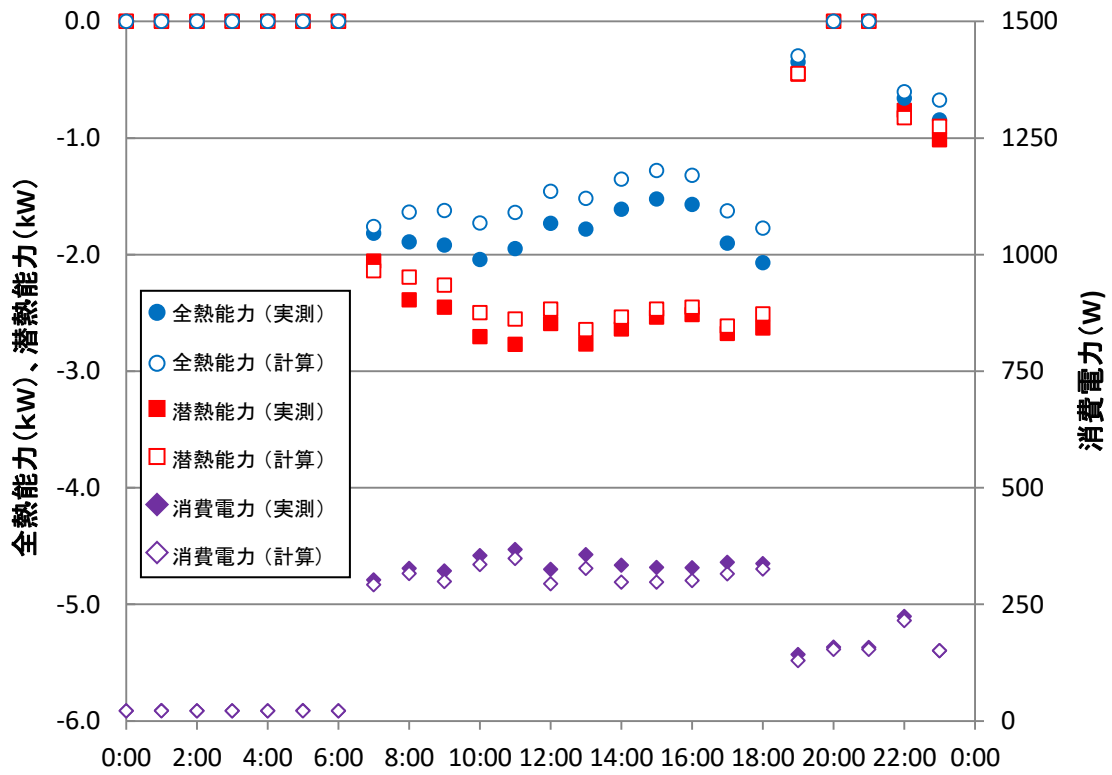


図 28 実測結果，計算結果比較 (7/8)

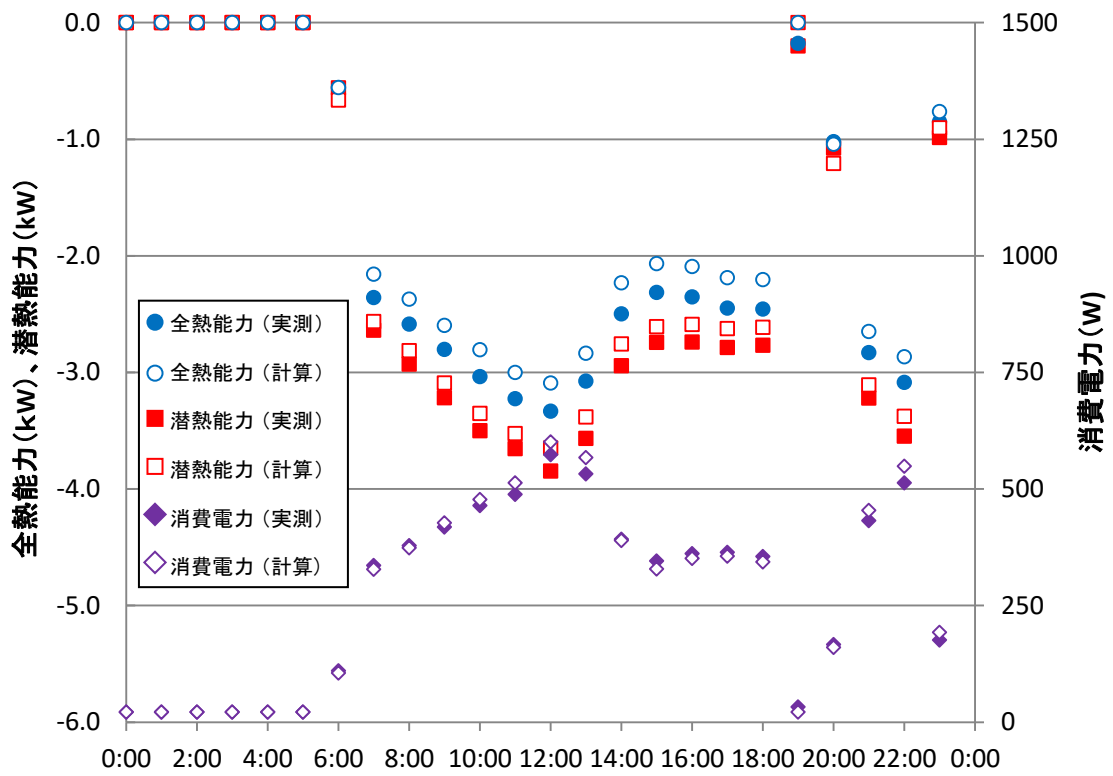


図 29 実測結果，計算結果比較 (7/9)

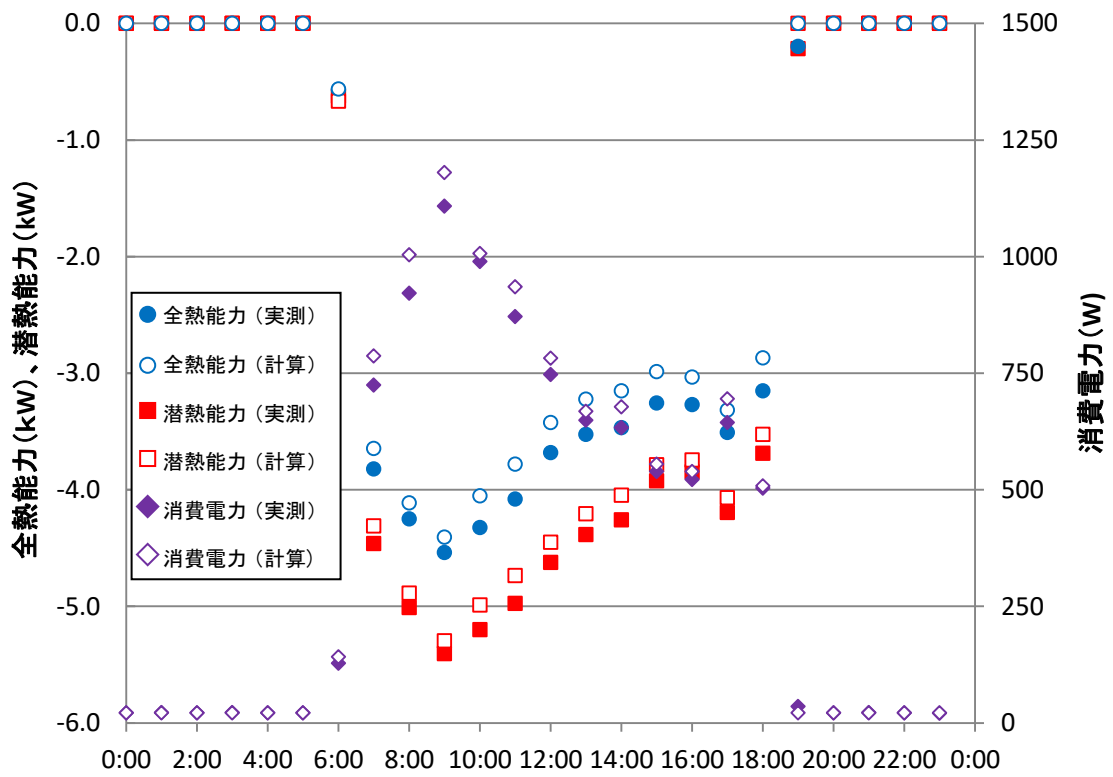


図 30 実測結果，計算結果比較 (7/10)

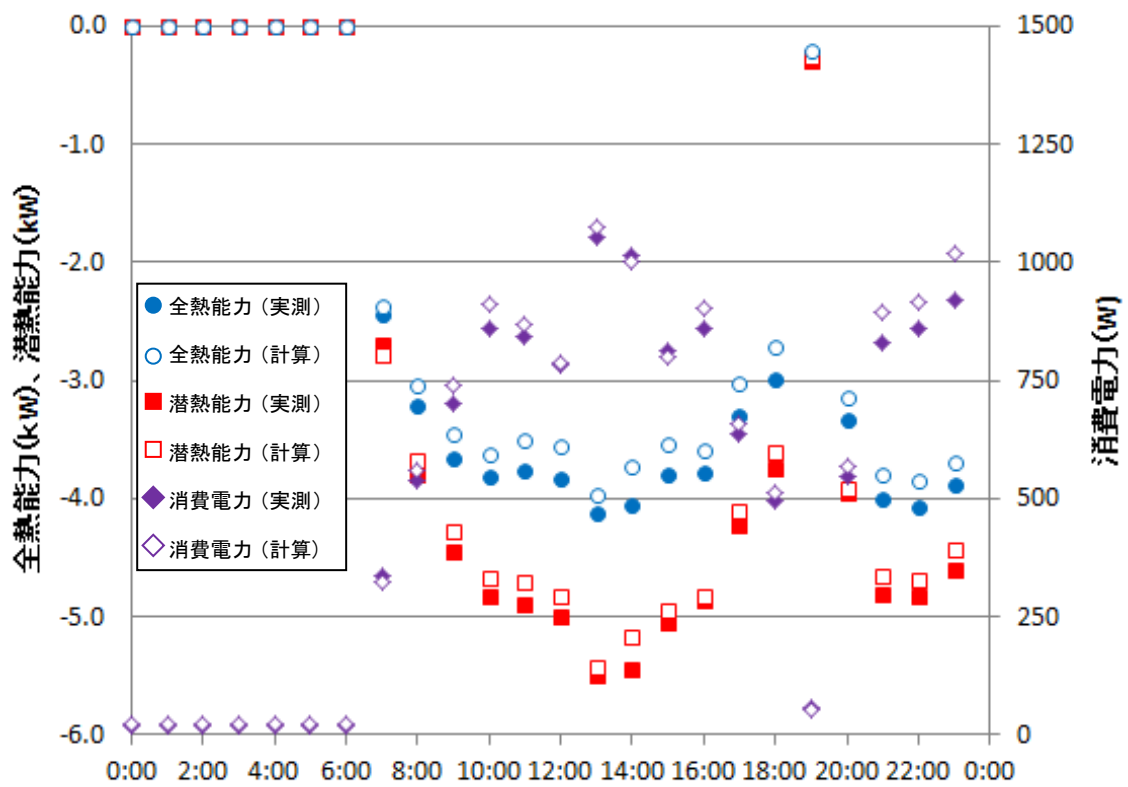


図 31 実測結果，計算結果比較 (7/11)

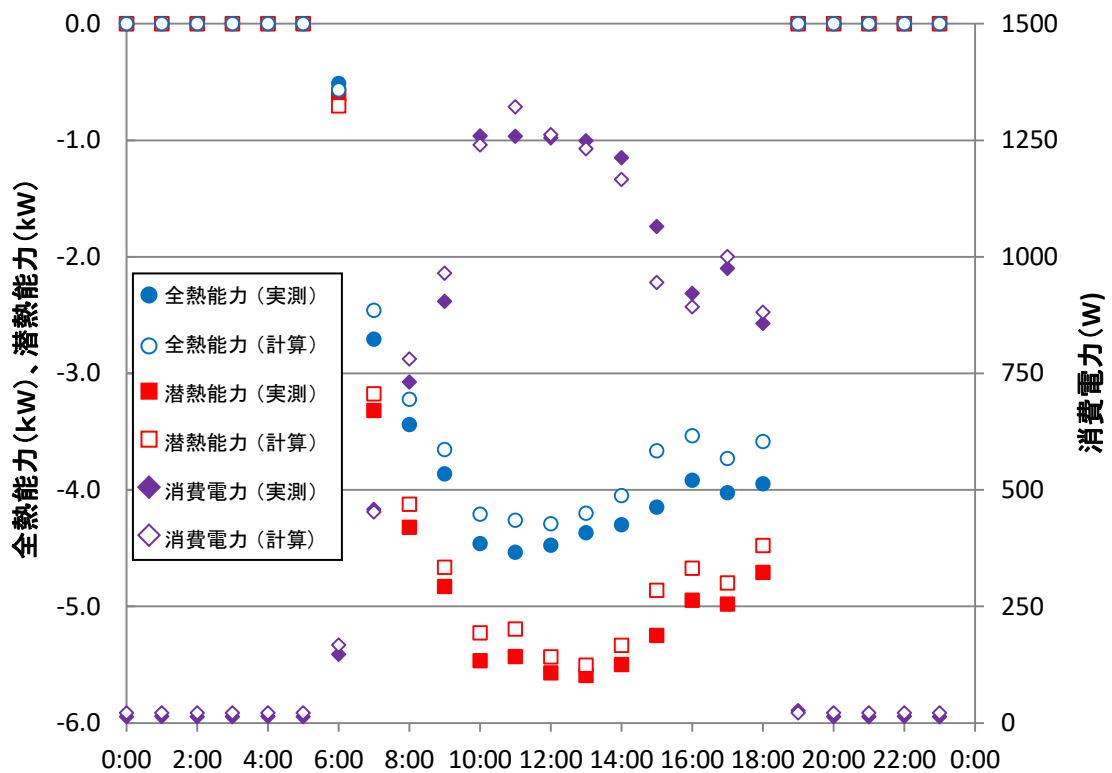


図 32 実測結果，計算結果比較 (7/14)

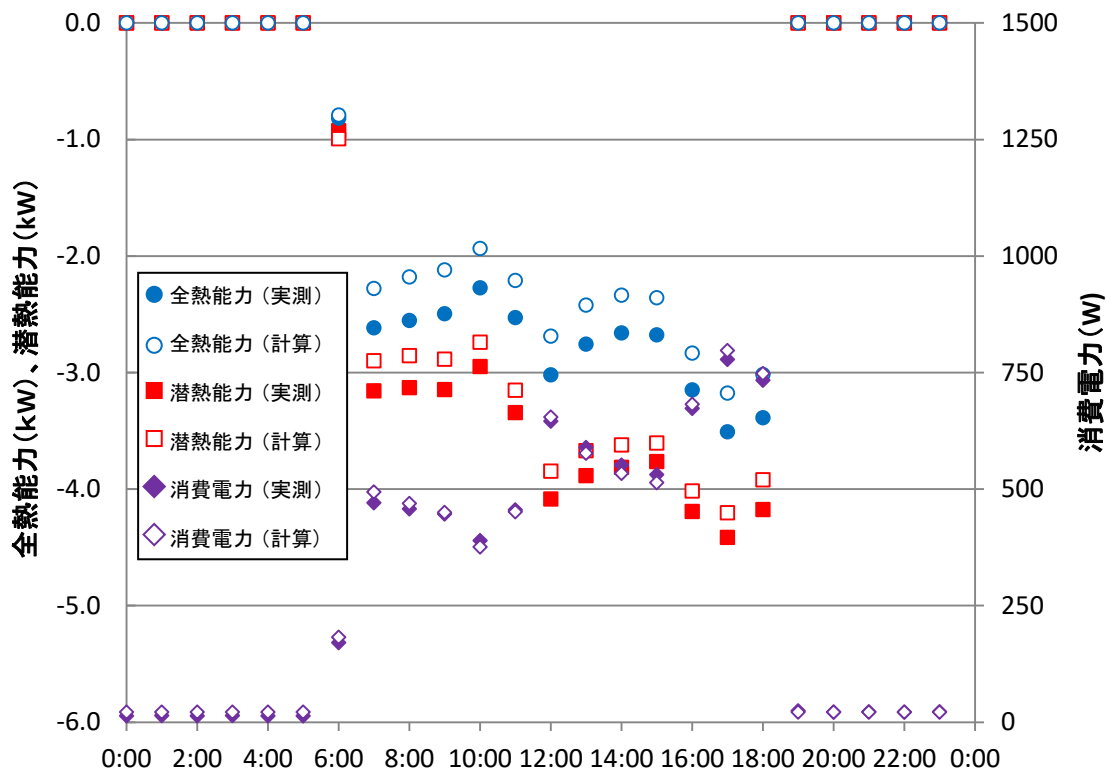


図 33 実測結果，計算結果比較 (7/15)

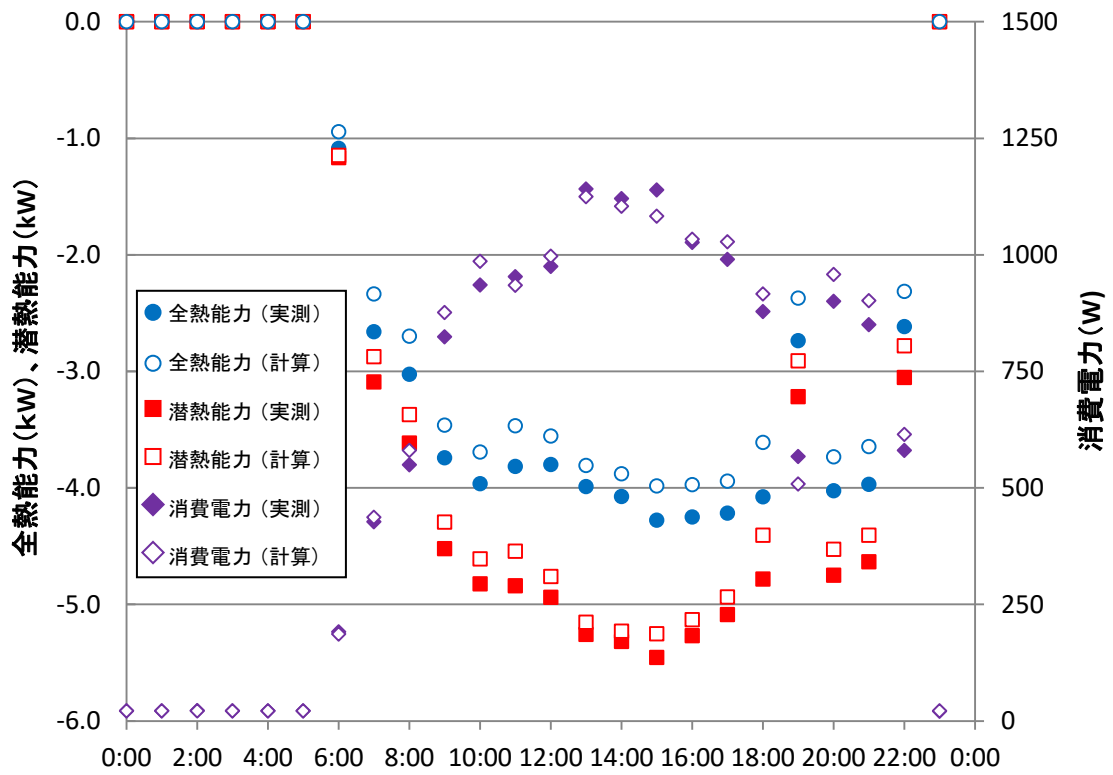


図 34 実測結果，計算結果比較 (7/16)

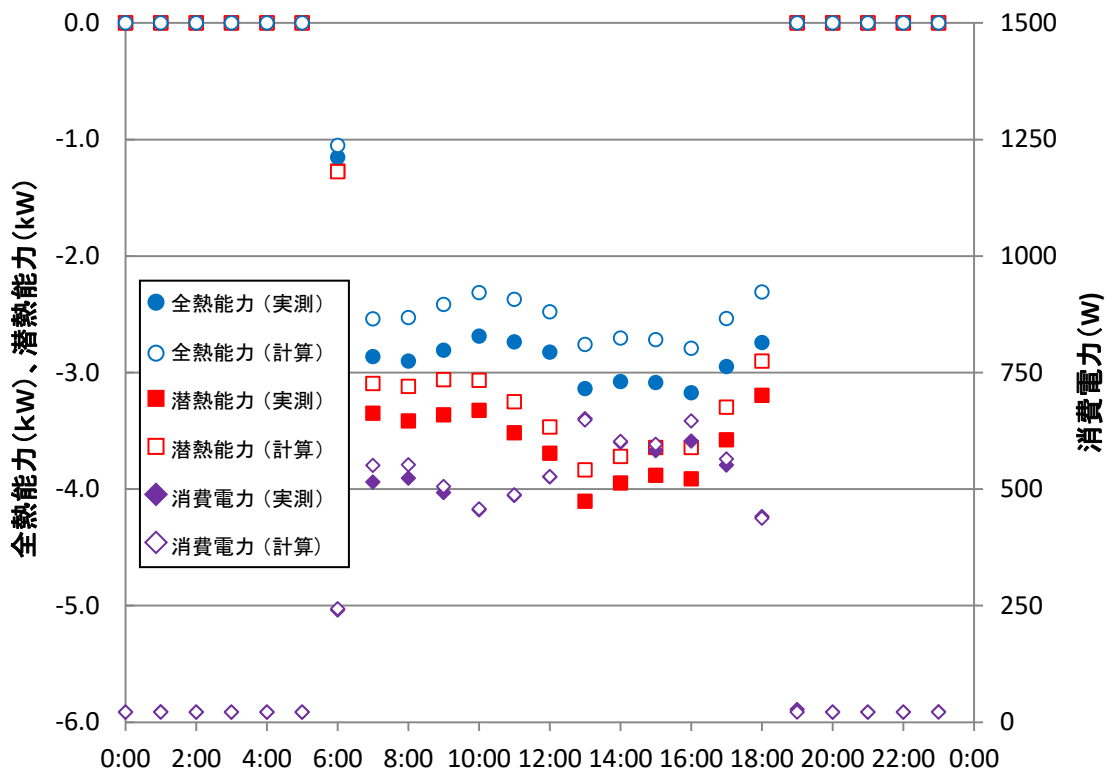


図 35 実測結果，計算結果比較 (7/17)

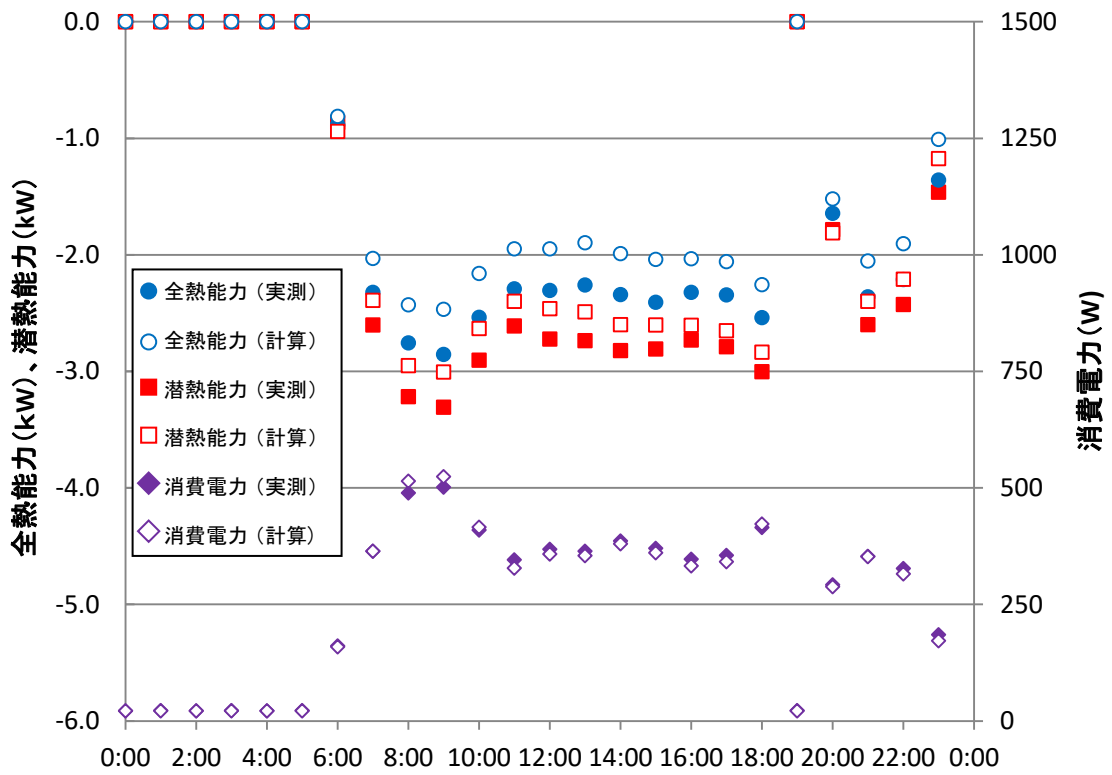


図 36 実測結果，計算結果比較 (7/18)

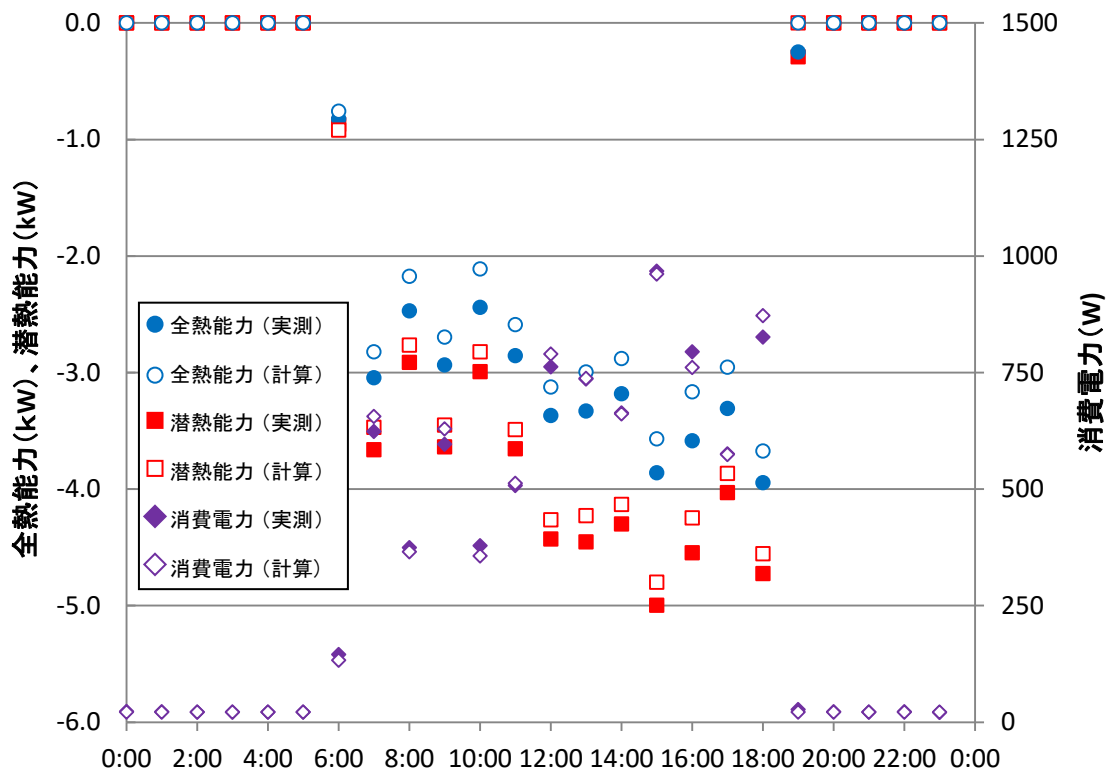


図 37 実測結果, 計算結果比較 (7/22)

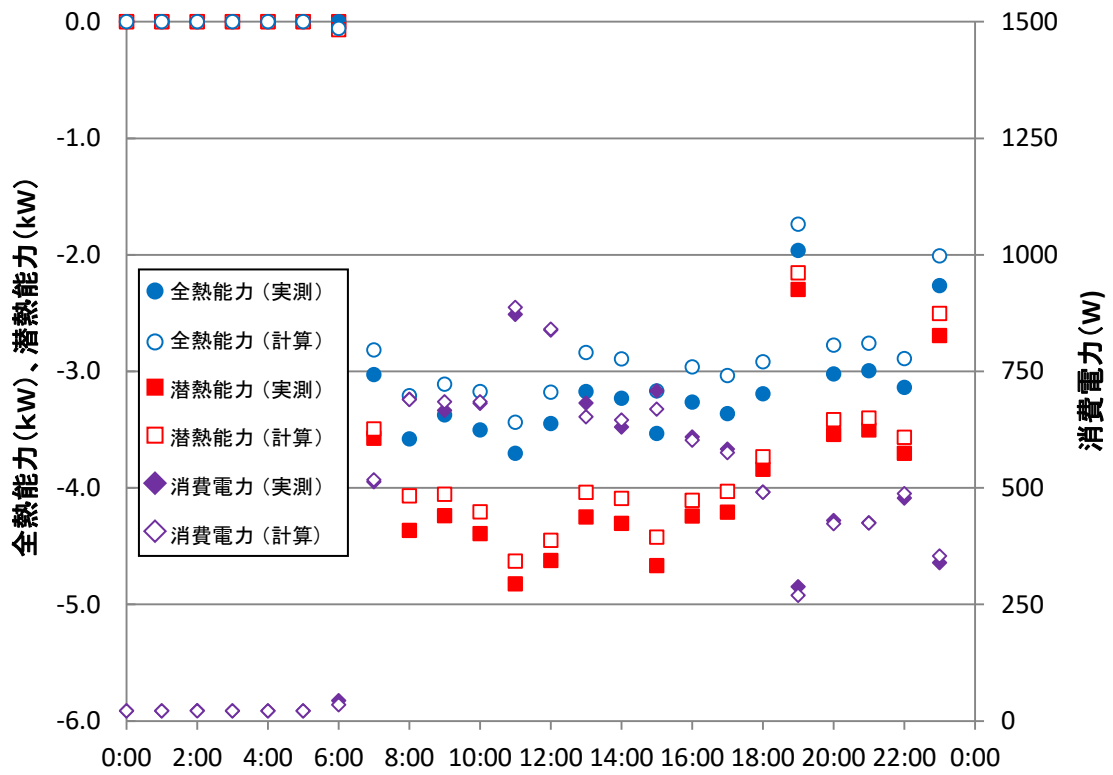


図 38 実測結果, 計算結果比較 (7/23)

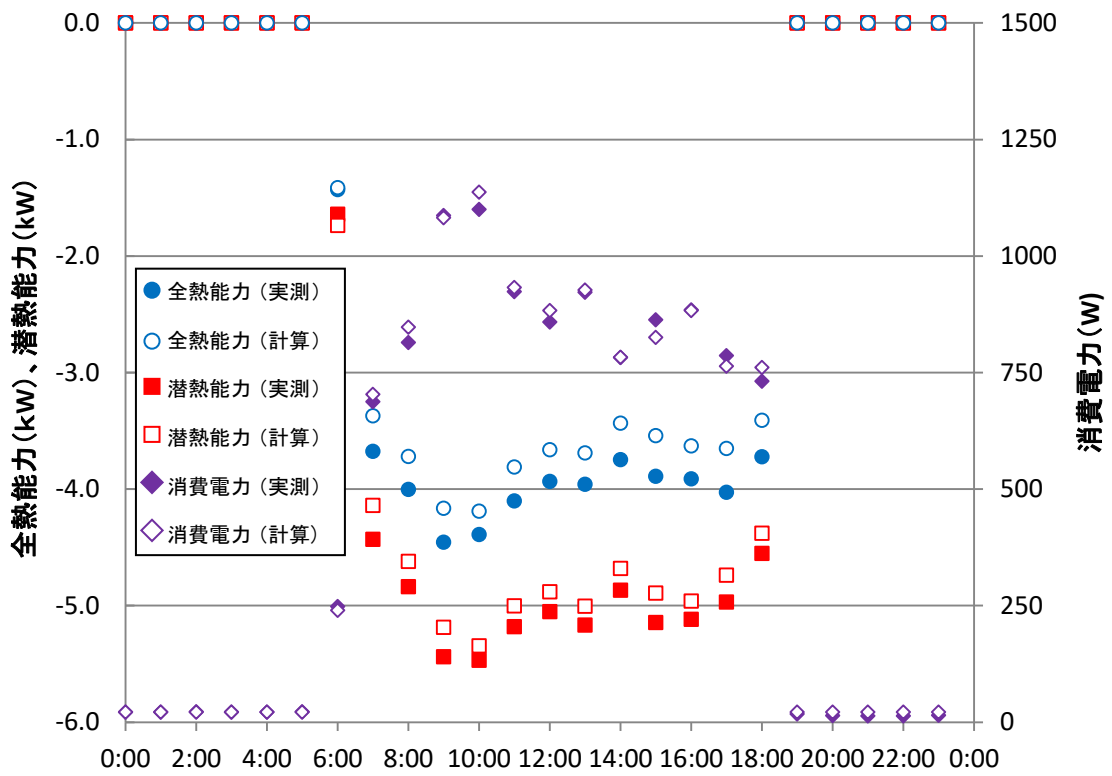


図 39 実測結果, 計算結果比較 (7/24)

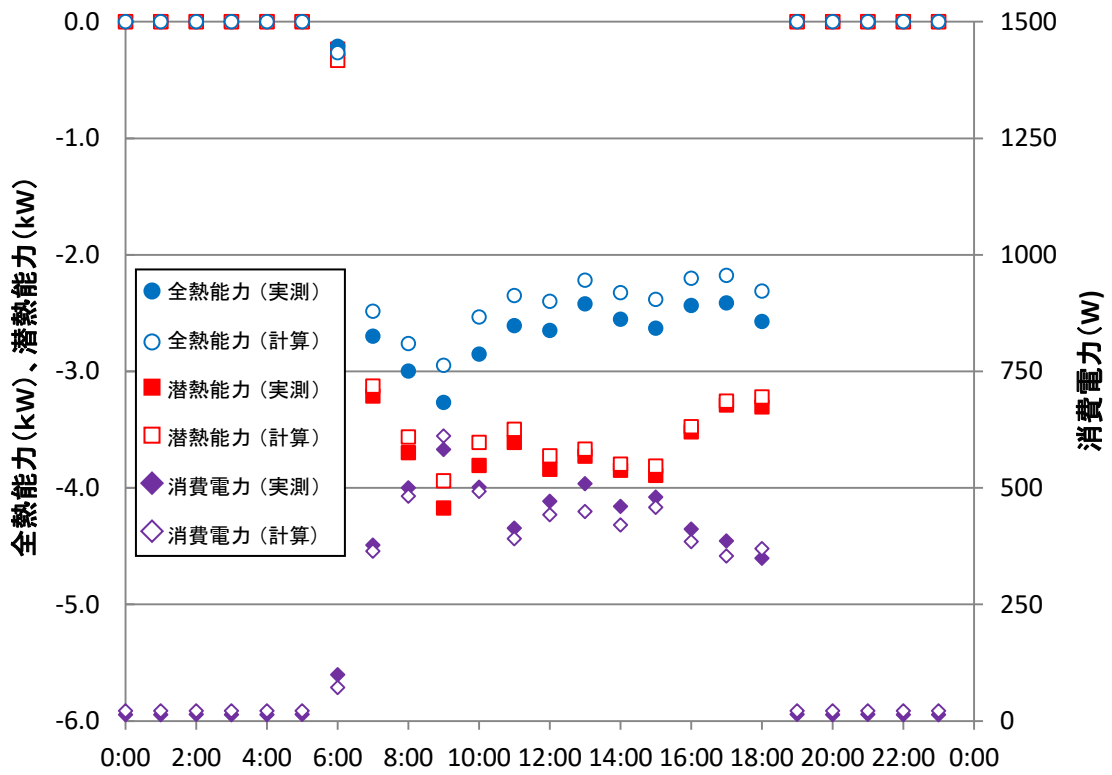


図 40 実測結果, 計算結果比較 (7/25)

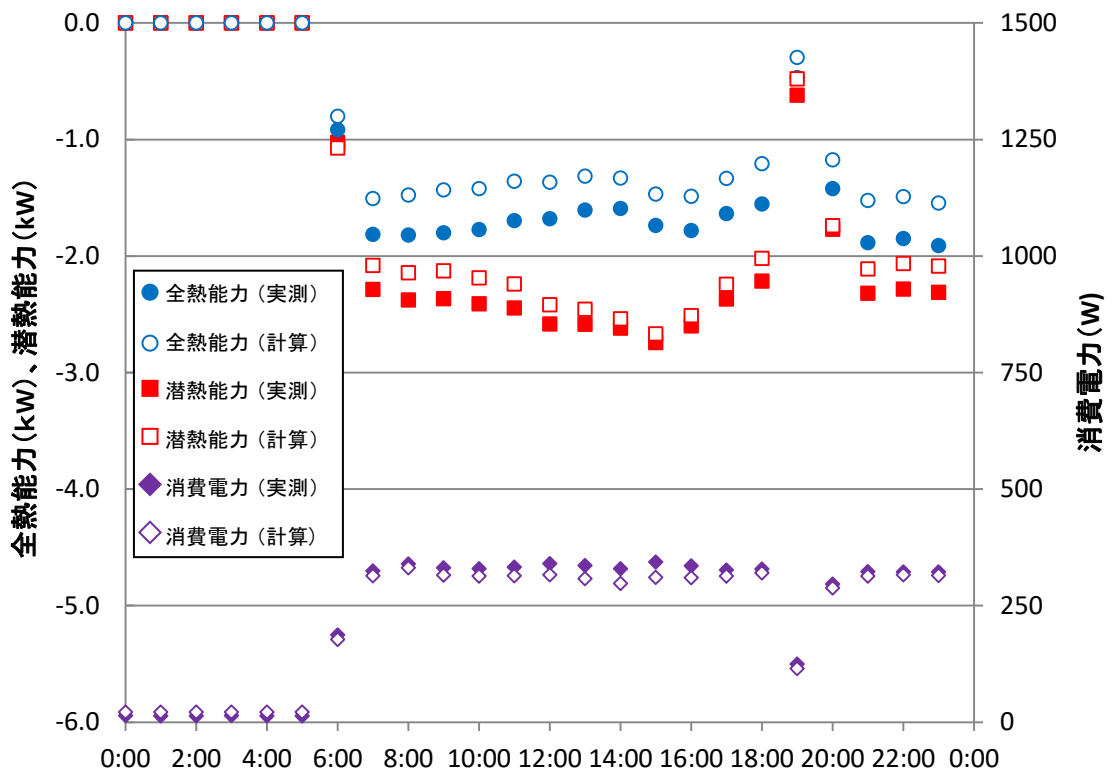


圖 41 實測結果，計算結果比較 (7/28)

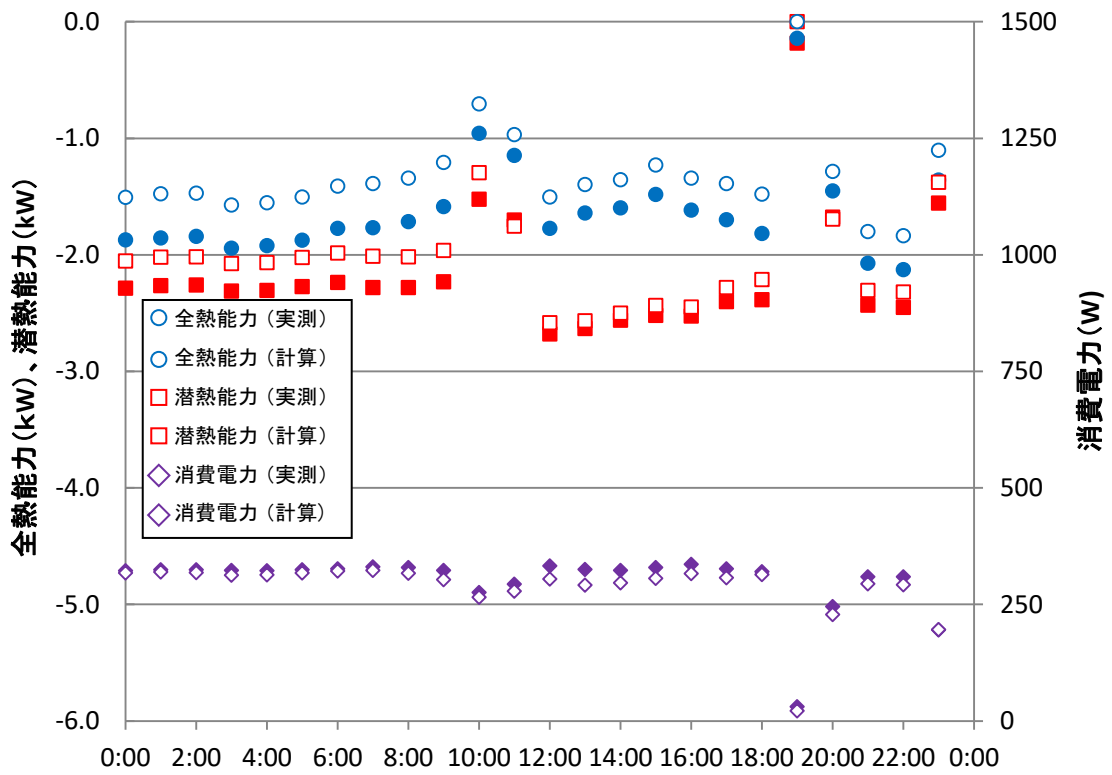


圖 42 實測結果，計算結果比較 (7/29)

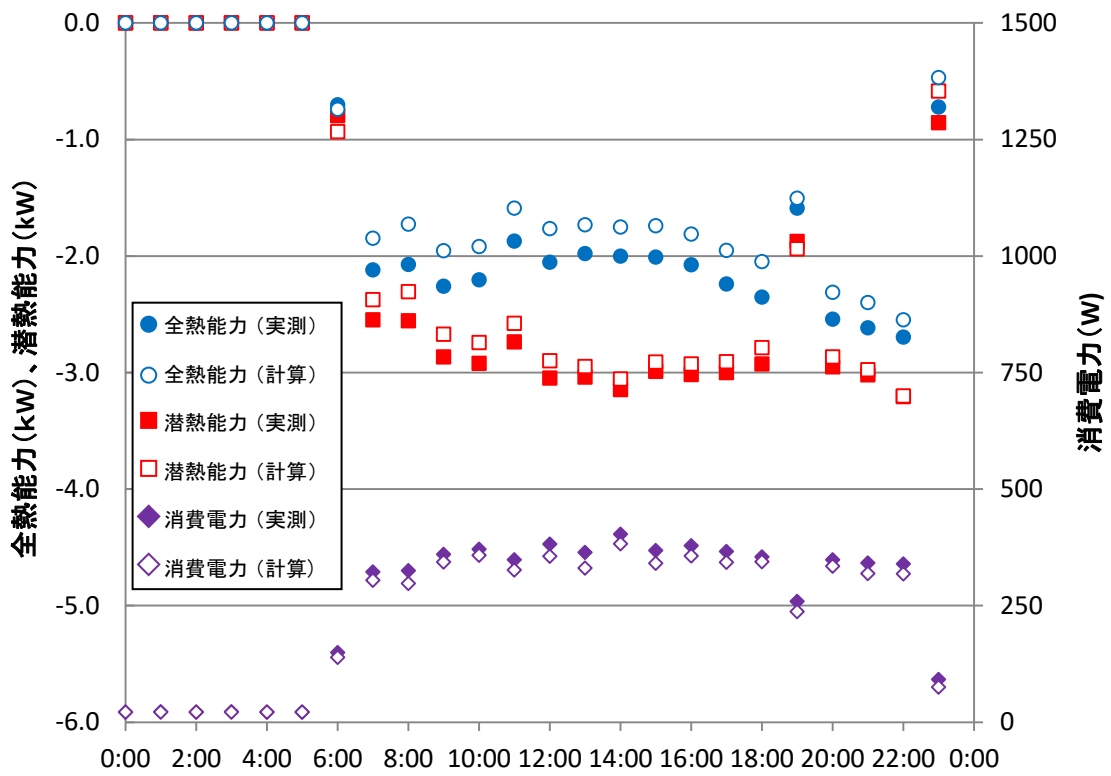


圖 43 實測結果，計算結果比較 (7/30)

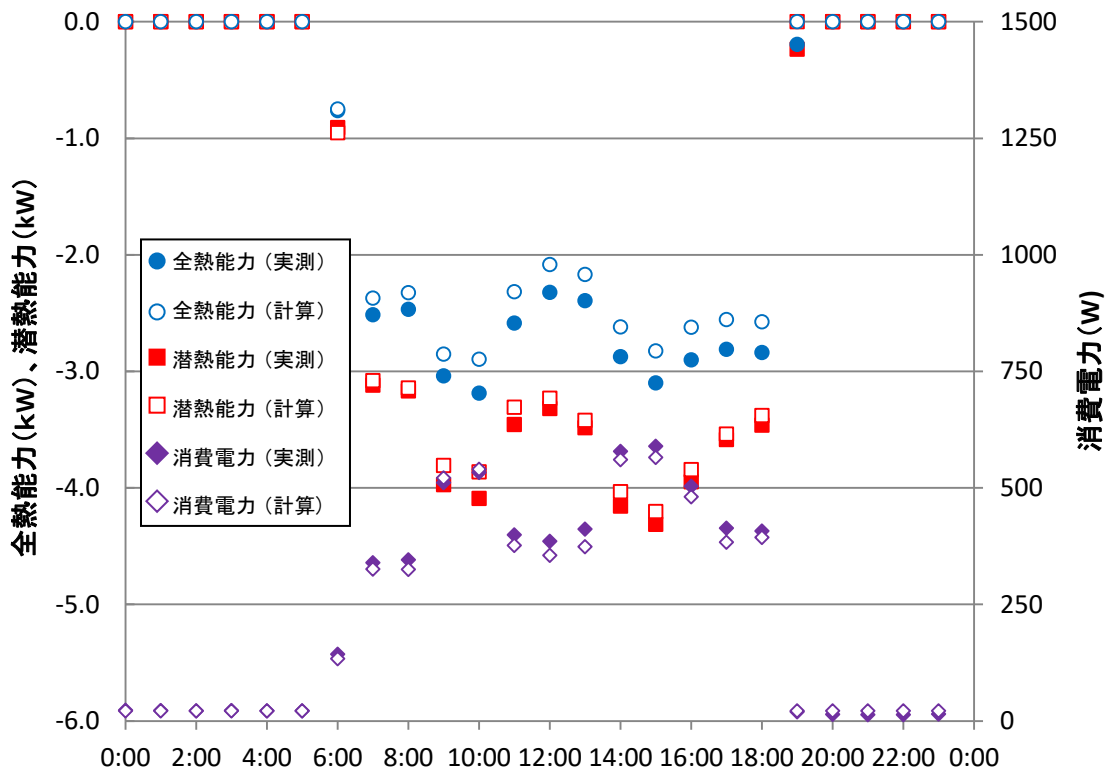


圖 44 實測結果，計算結果比較 (7/31)

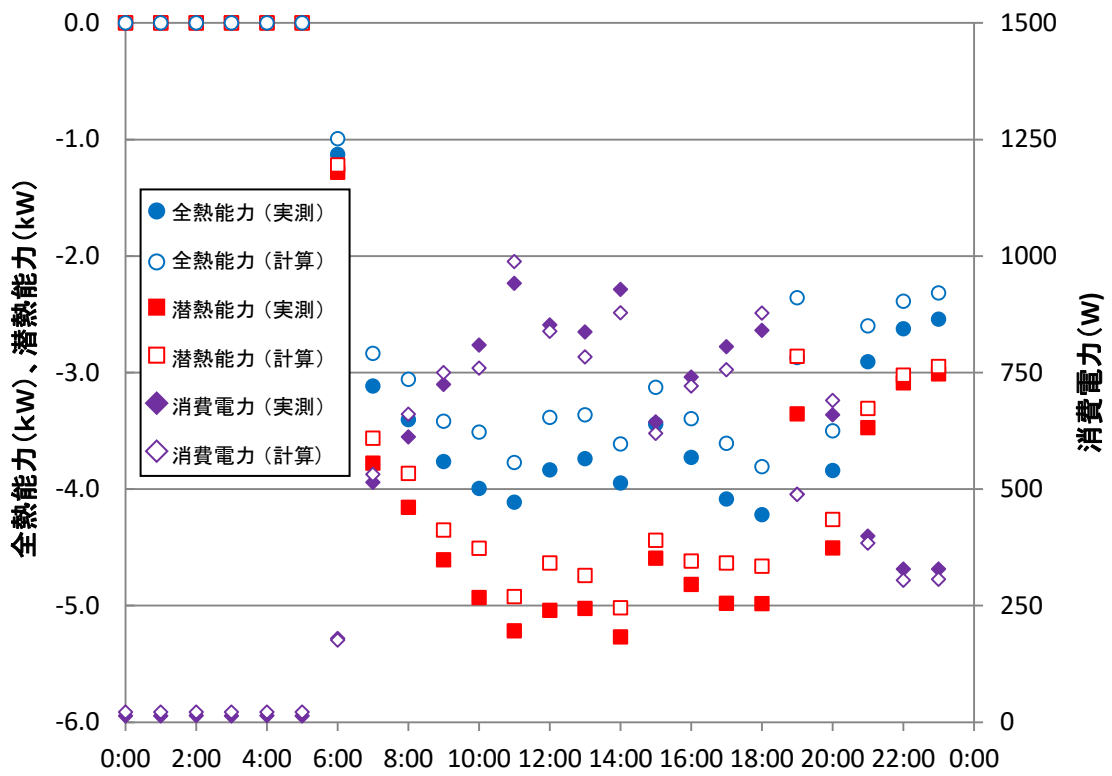


圖 45 實測結果，計算結果比較 (8/1)

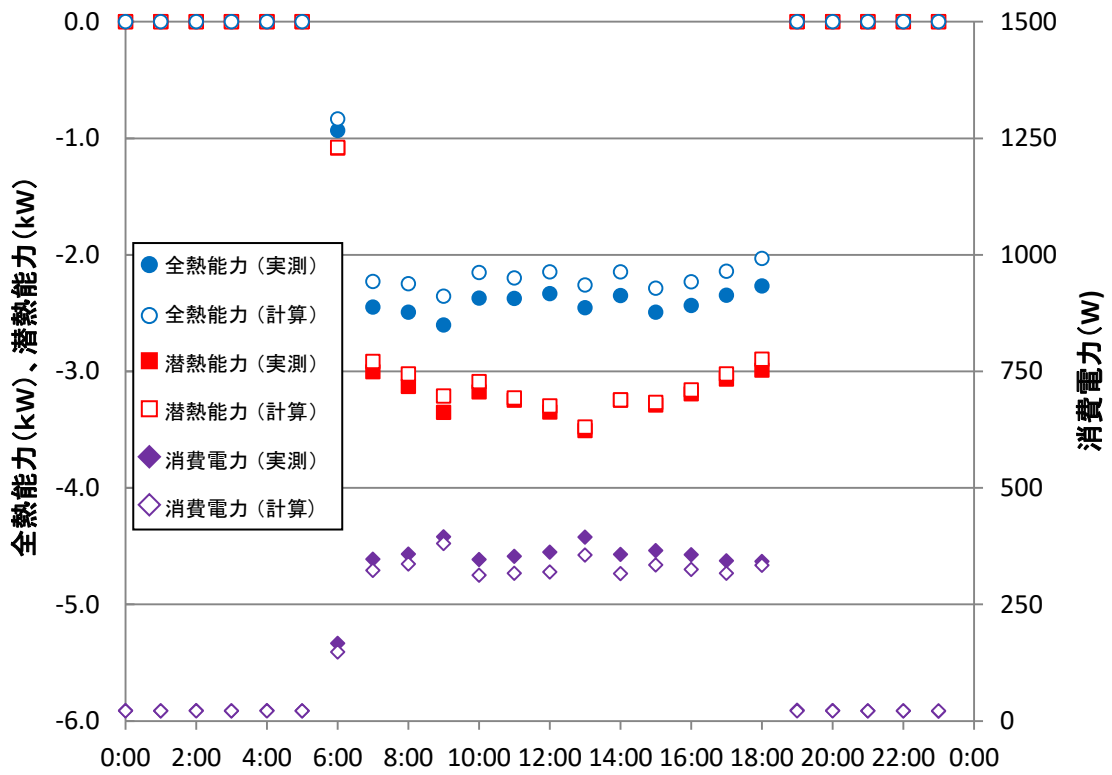


圖 46 實測結果，計算結果比較 (8/4)

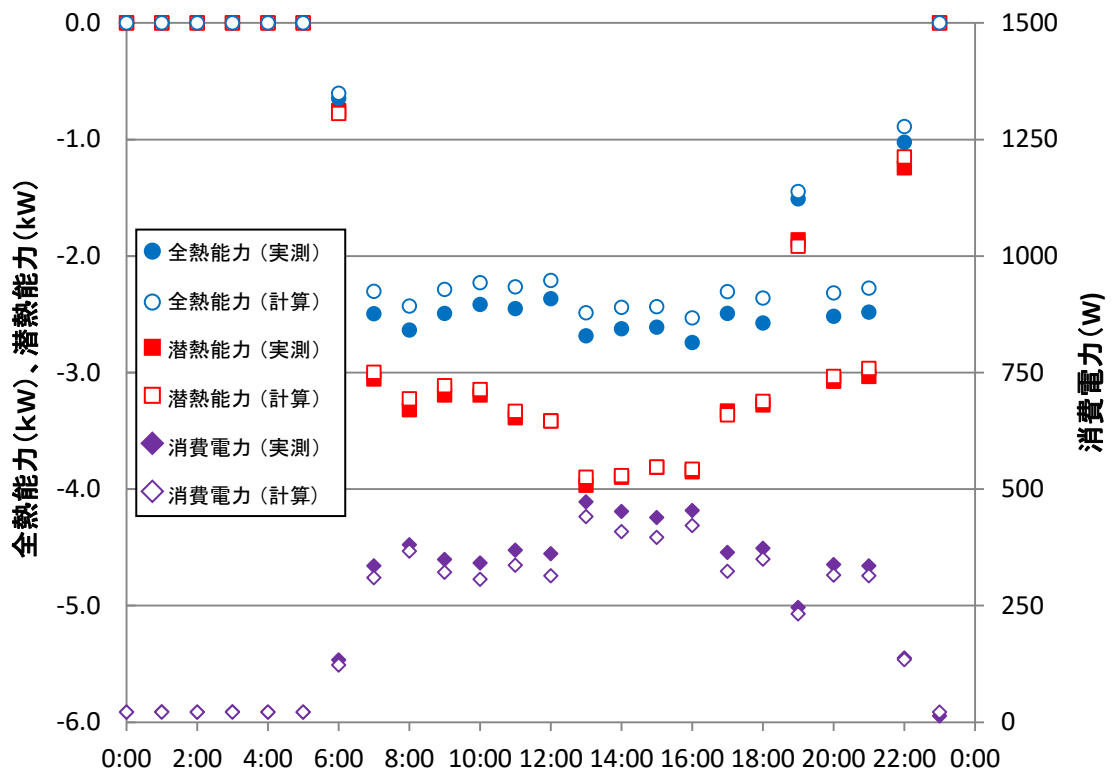


図 47 実測結果，計算結果比較 (8/5)

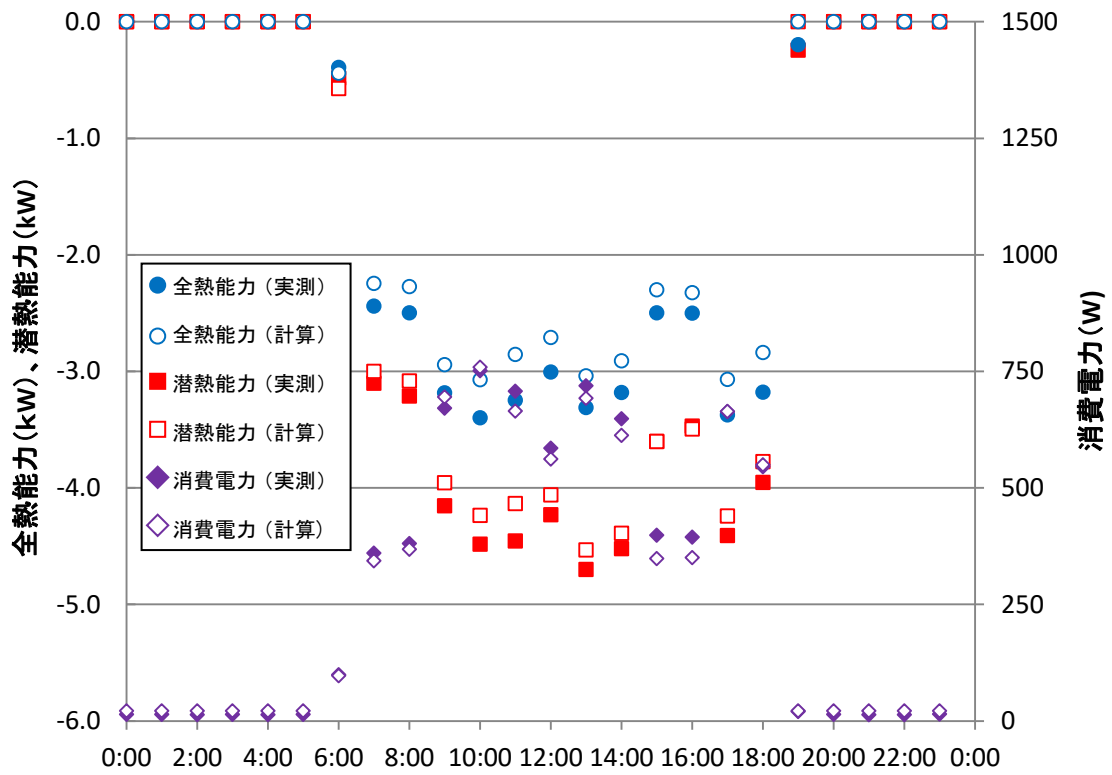


図 48 実測結果，計算結果比較 (8/6)

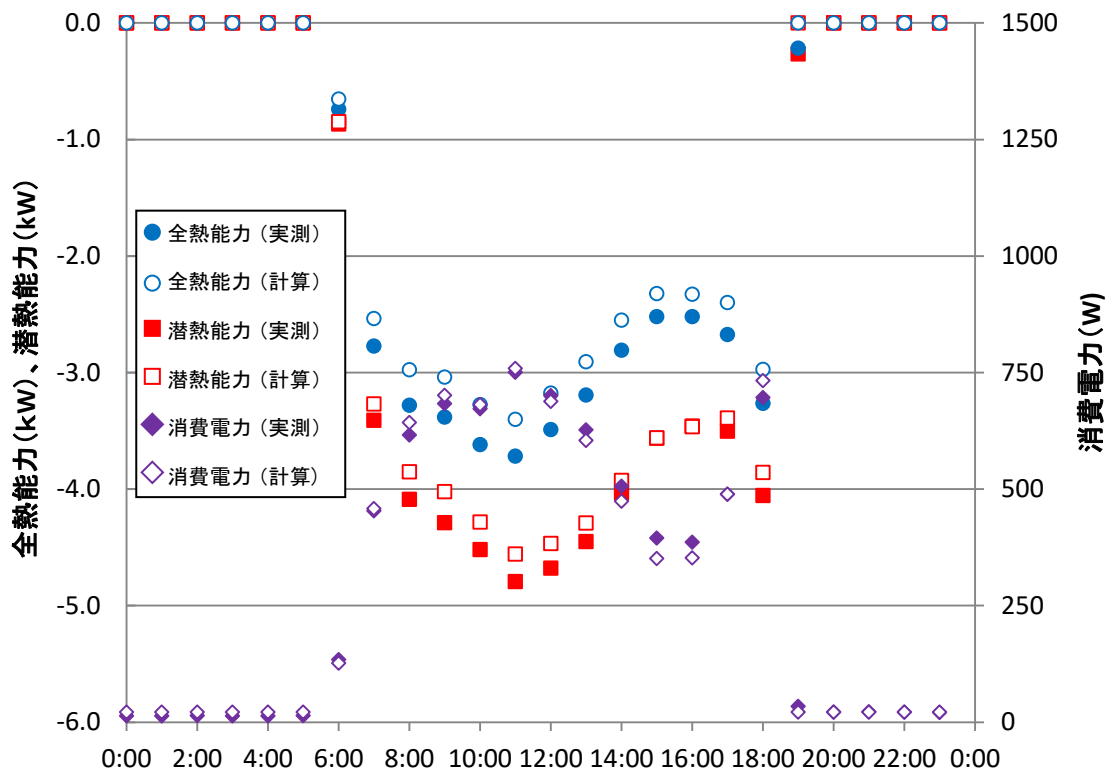


図 49 実測結果，計算結果比較 (8/7)

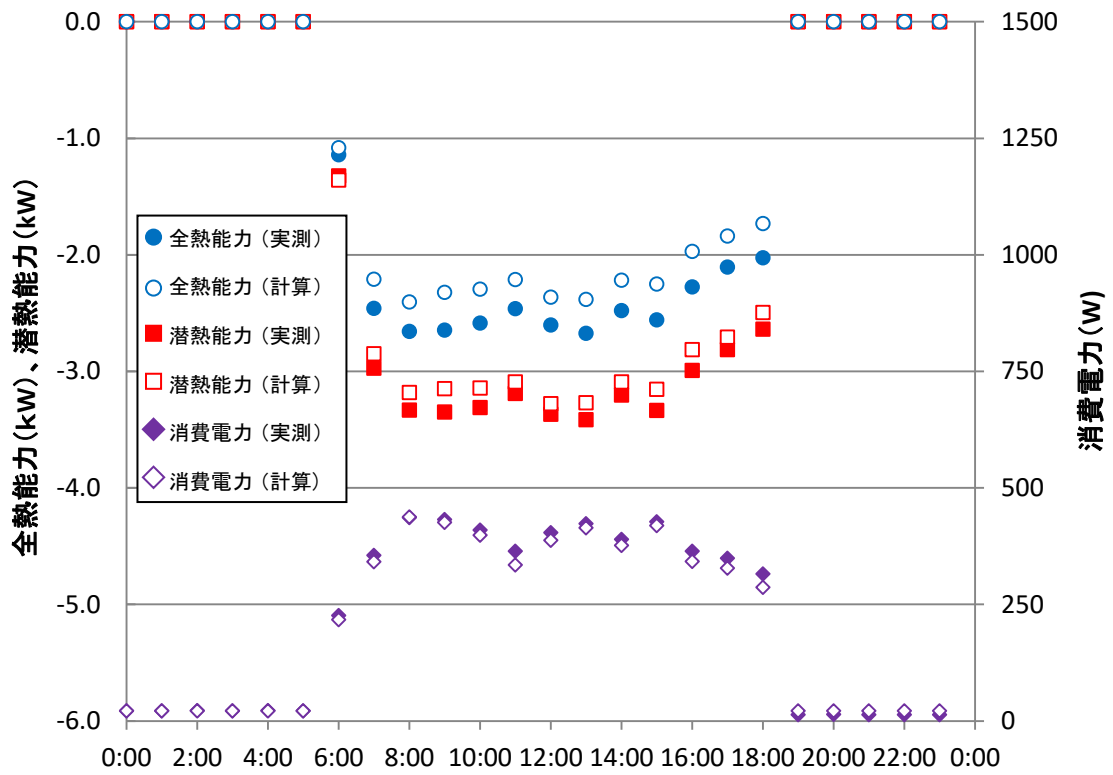


図 50 実測結果，計算結果比較 (8/8)

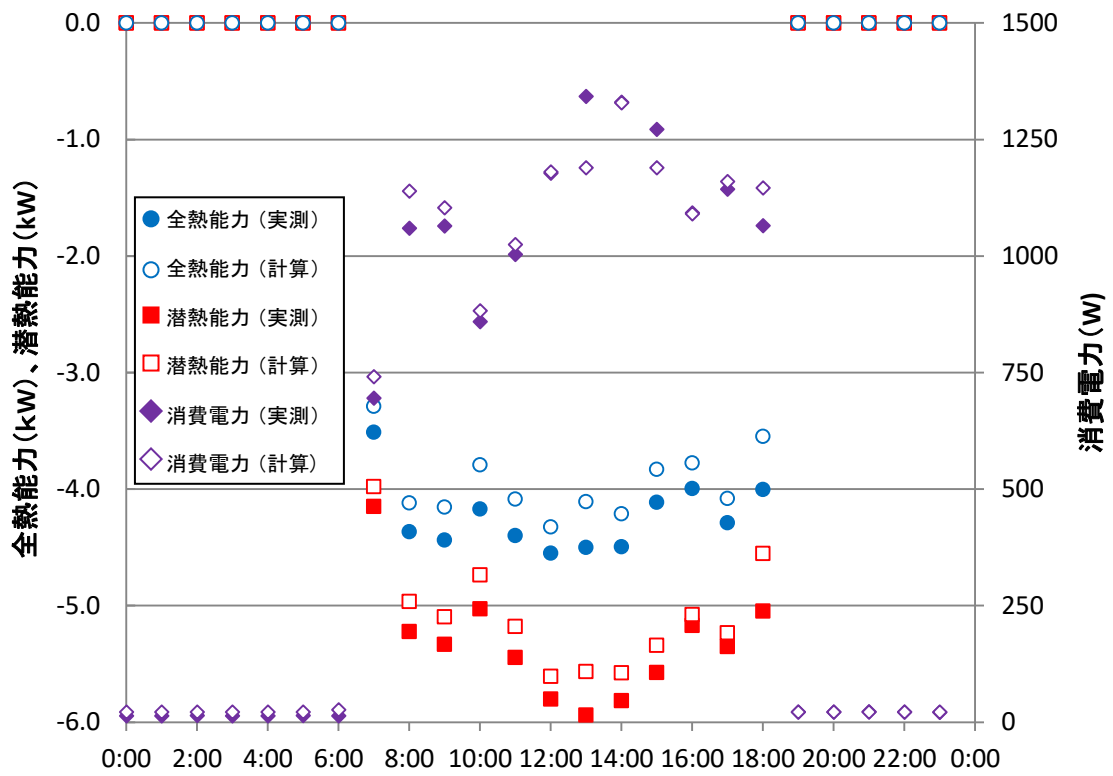


図 51 実測結果, 計算結果比較 (8/18)

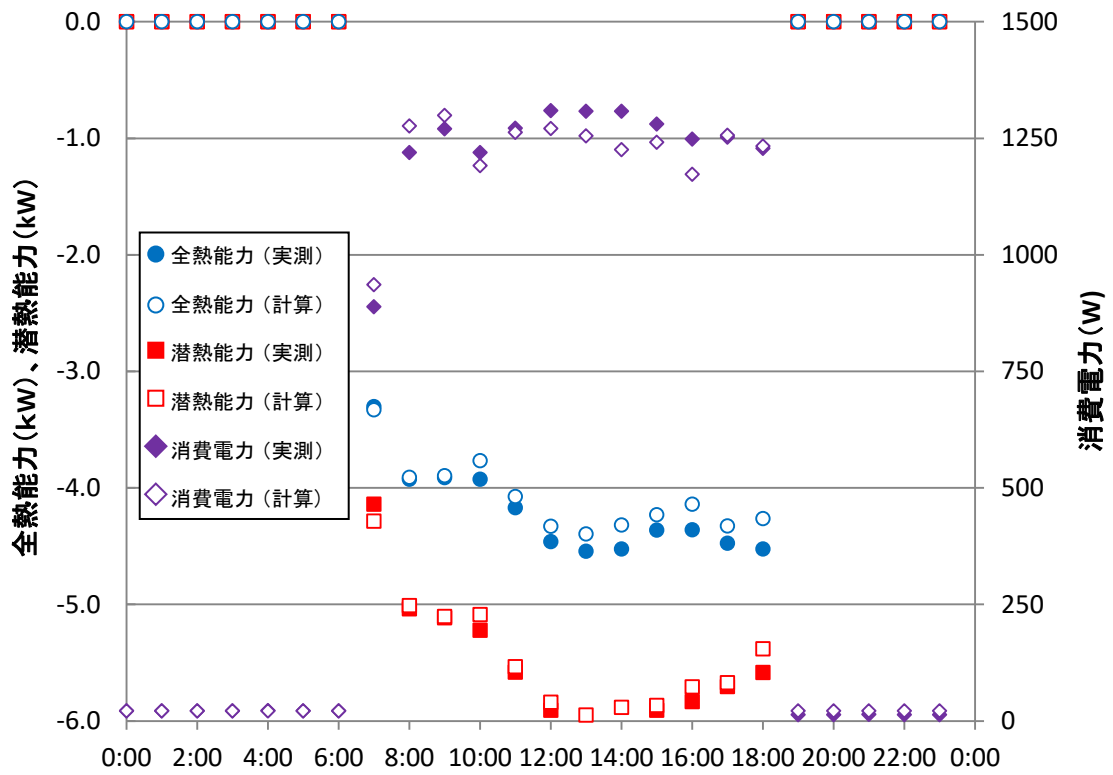


図 52 実測結果, 計算結果比較 (8/19)

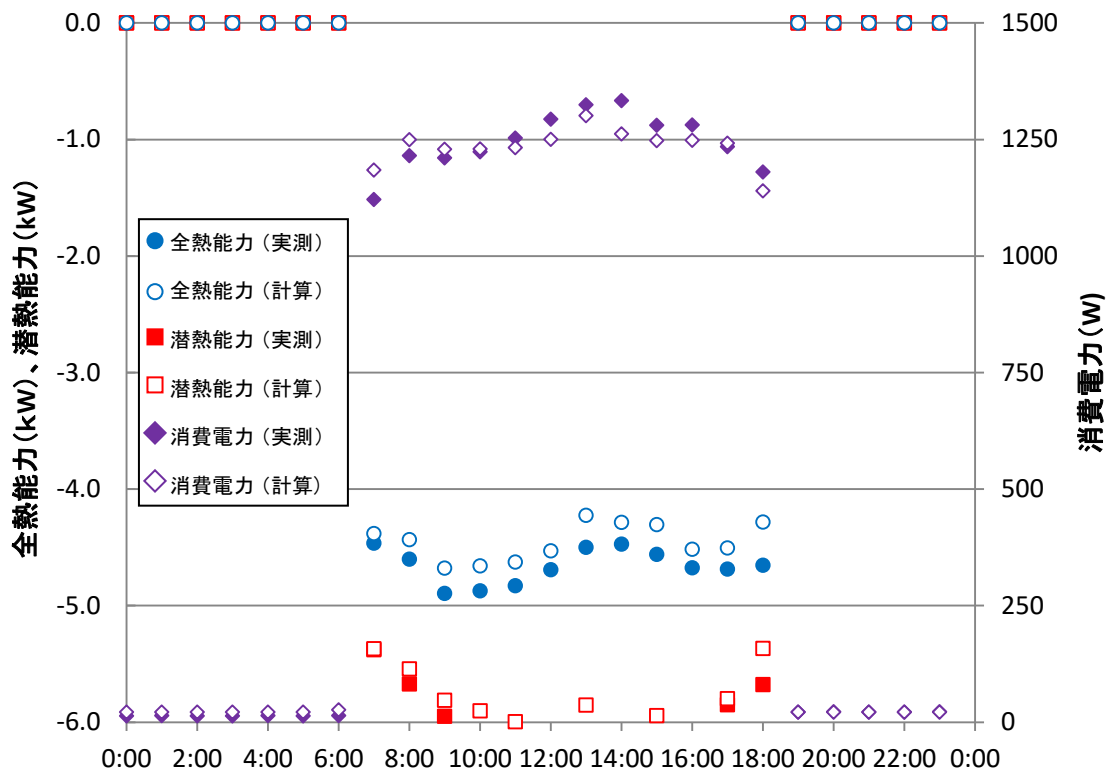


圖 53 實測結果，計算結果比較 (8/20)

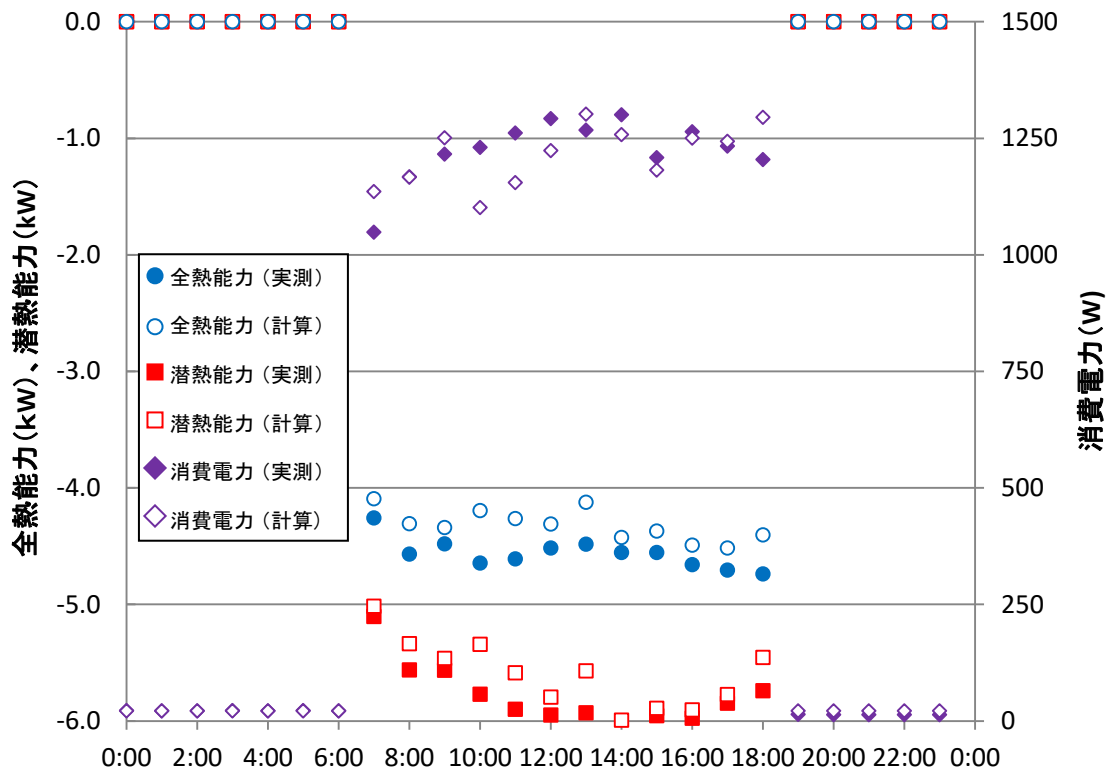


圖 54 實測結果，計算結果比較 (8/21)

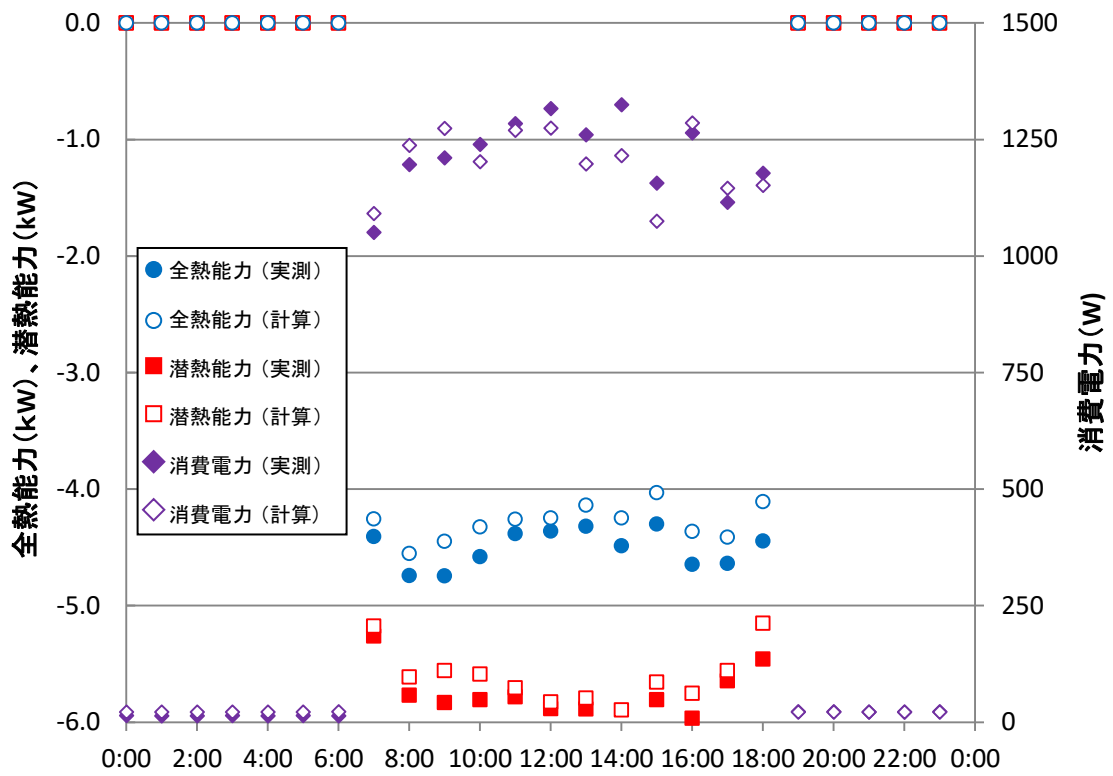


圖 55 實測結果，計算結果比較 (8/22)

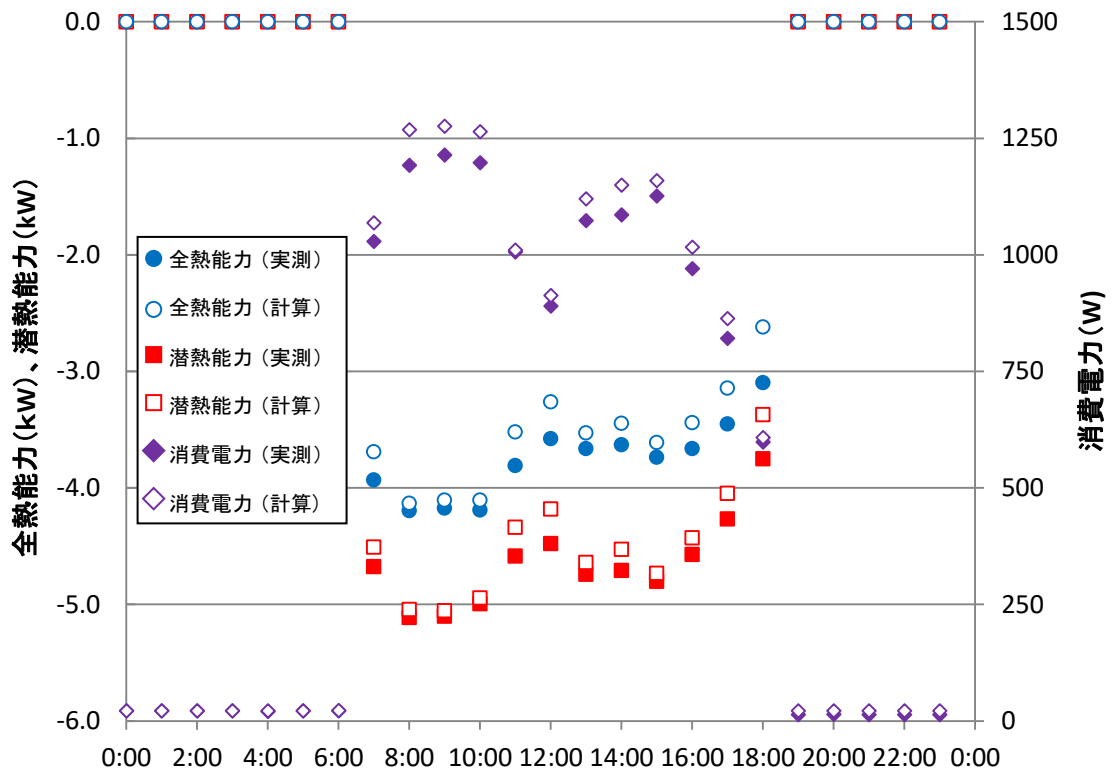


圖 56 實測結果，計算結果比較 (8/25)

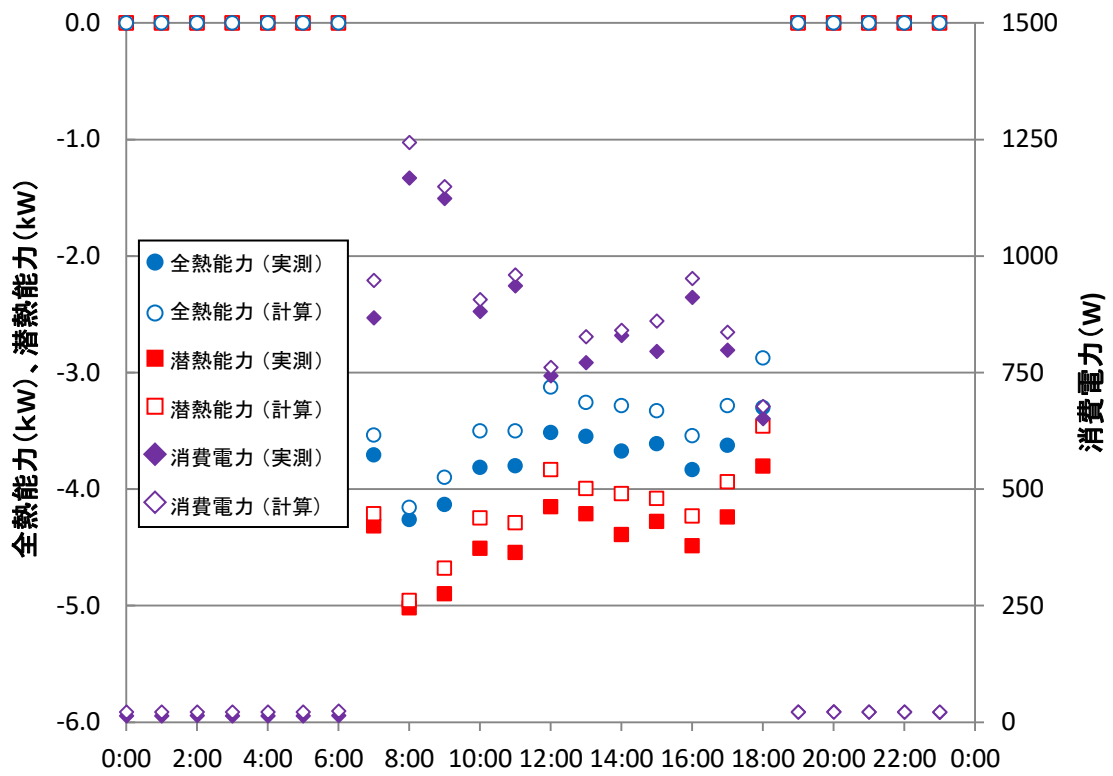


図 57 実測結果，計算結果比較 (8/26)

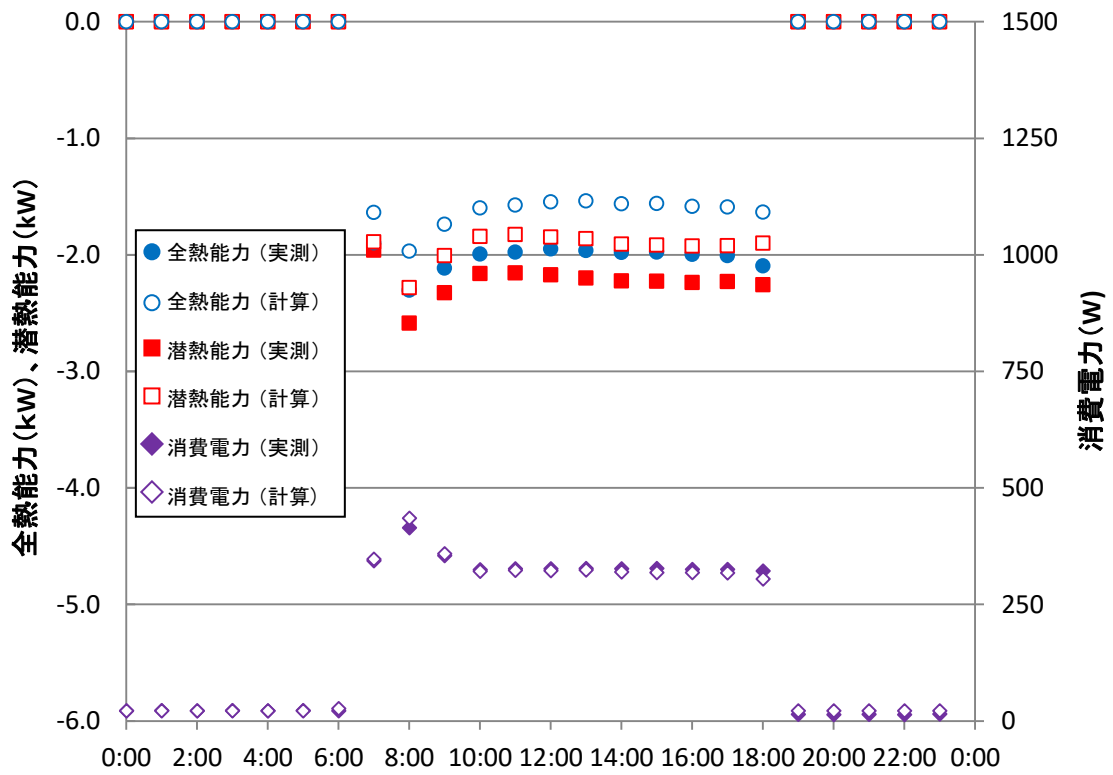


図 58 実測結果，計算結果比較 (8/27)

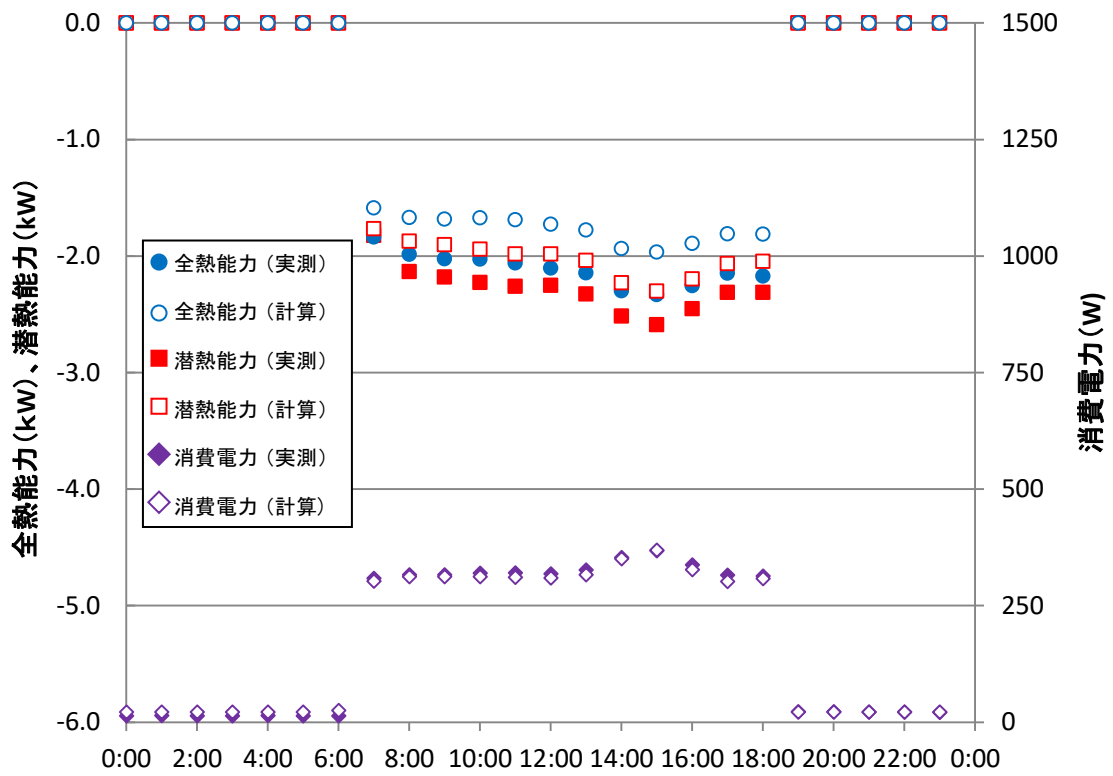


図 59 実測結果，計算結果比較 (8/28)

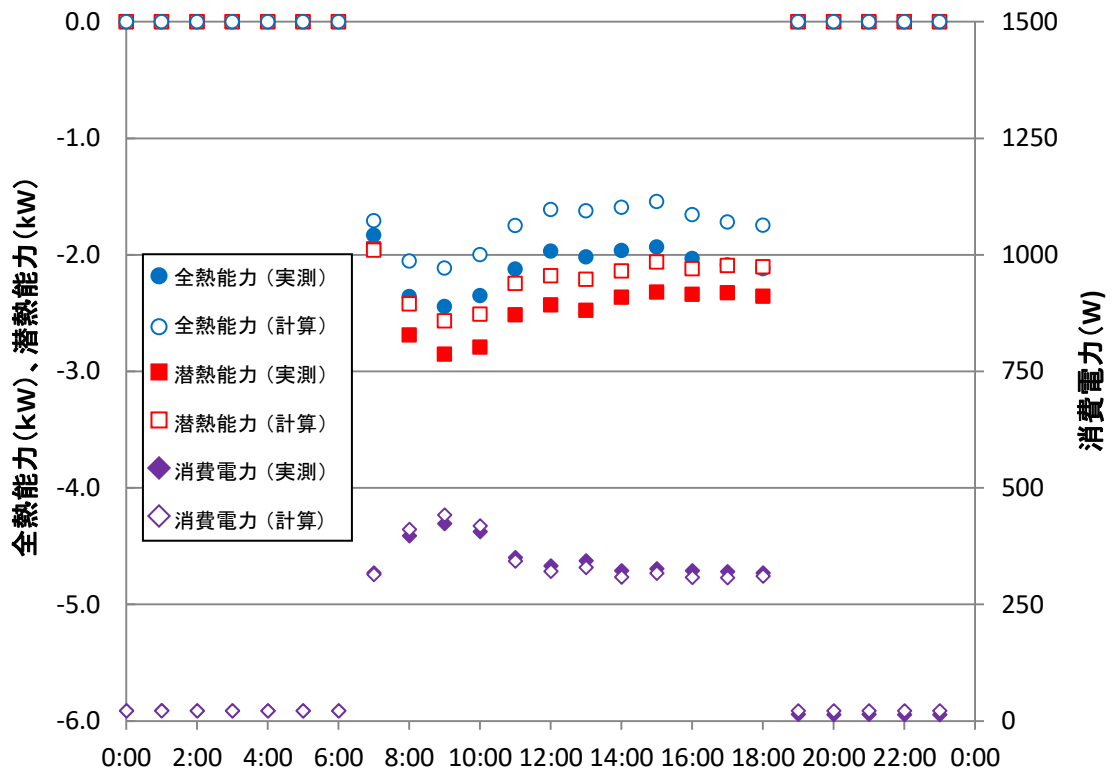


図 60 実測結果，計算結果比較 (8/29)

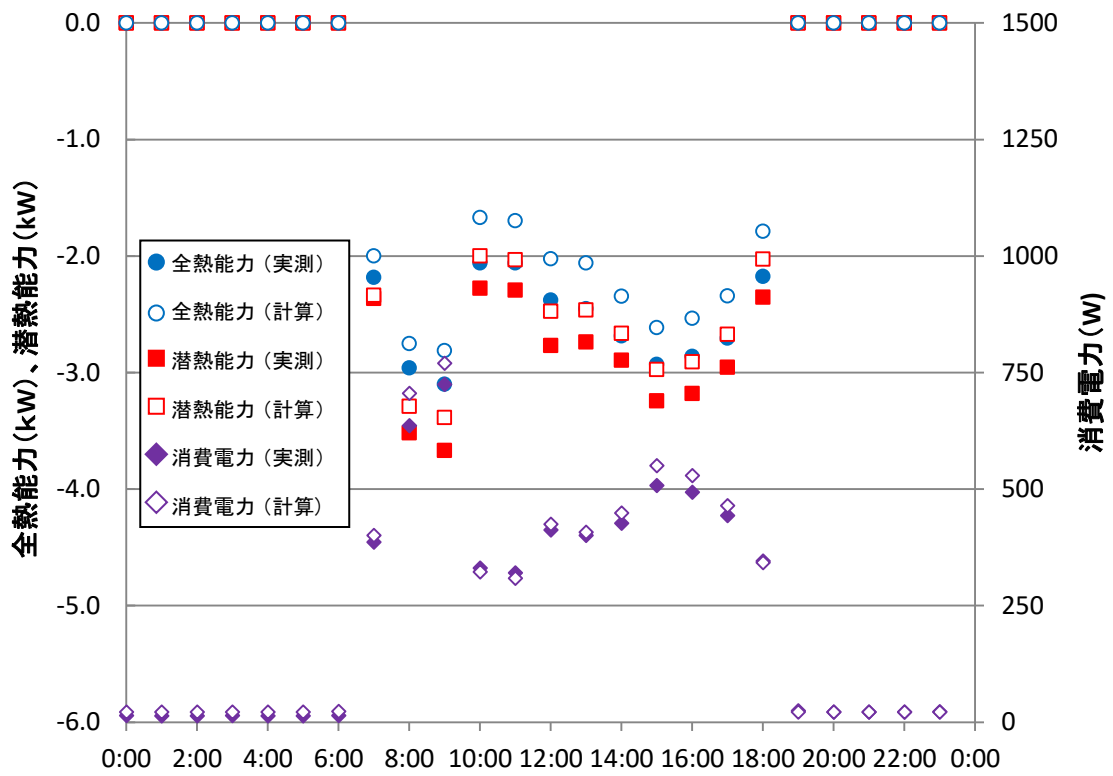


図 61 実測結果，計算結果比較 (9/1)

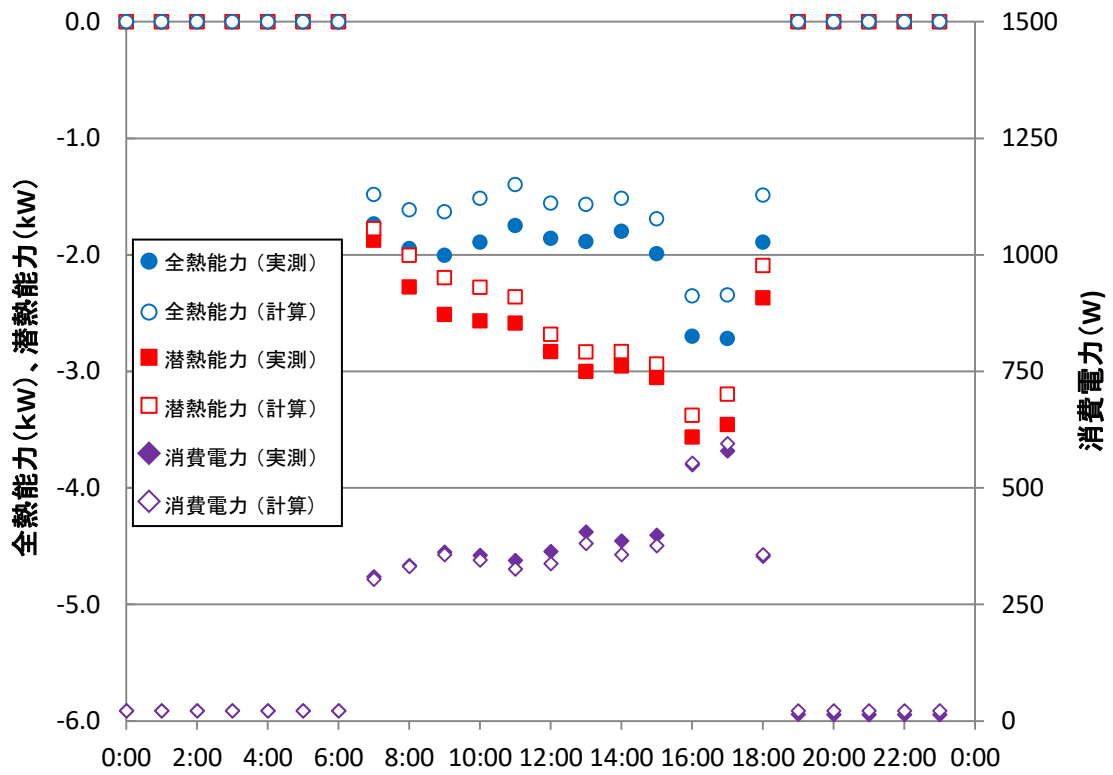


図 62 実測結果，計算結果比較 (9/2)

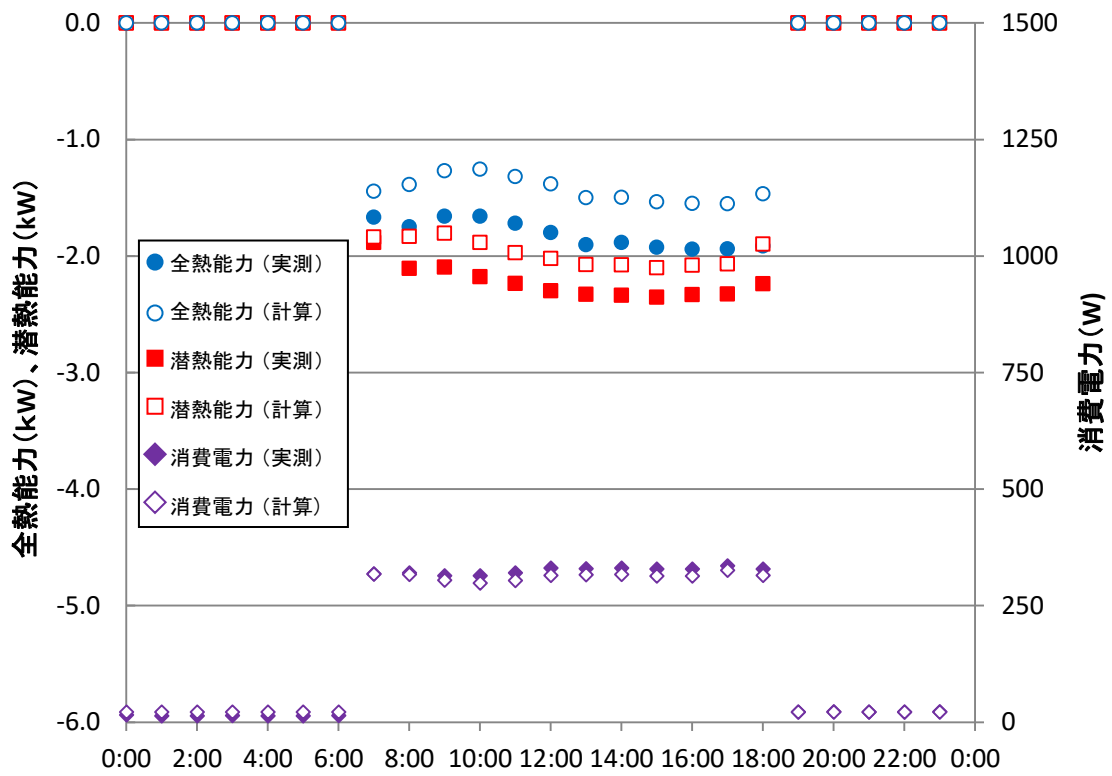


圖 63 實測結果，計算結果比較 (9/3)

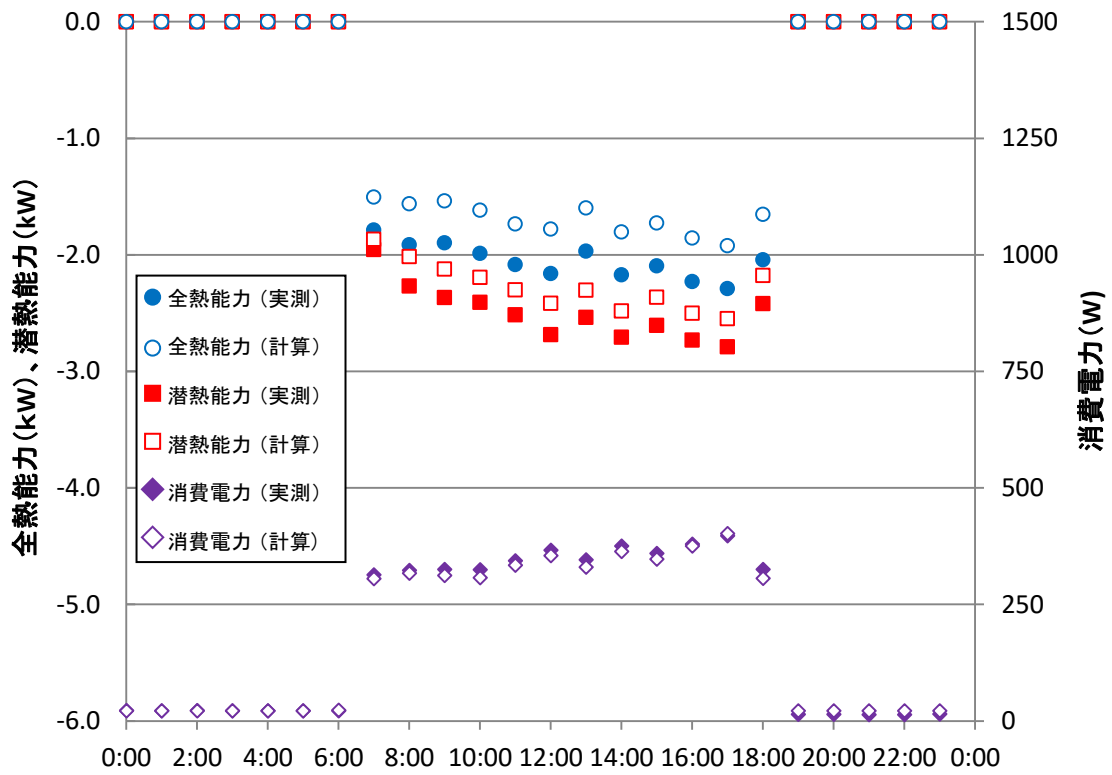


圖 64 實測結果，計算結果比較 (9/4)

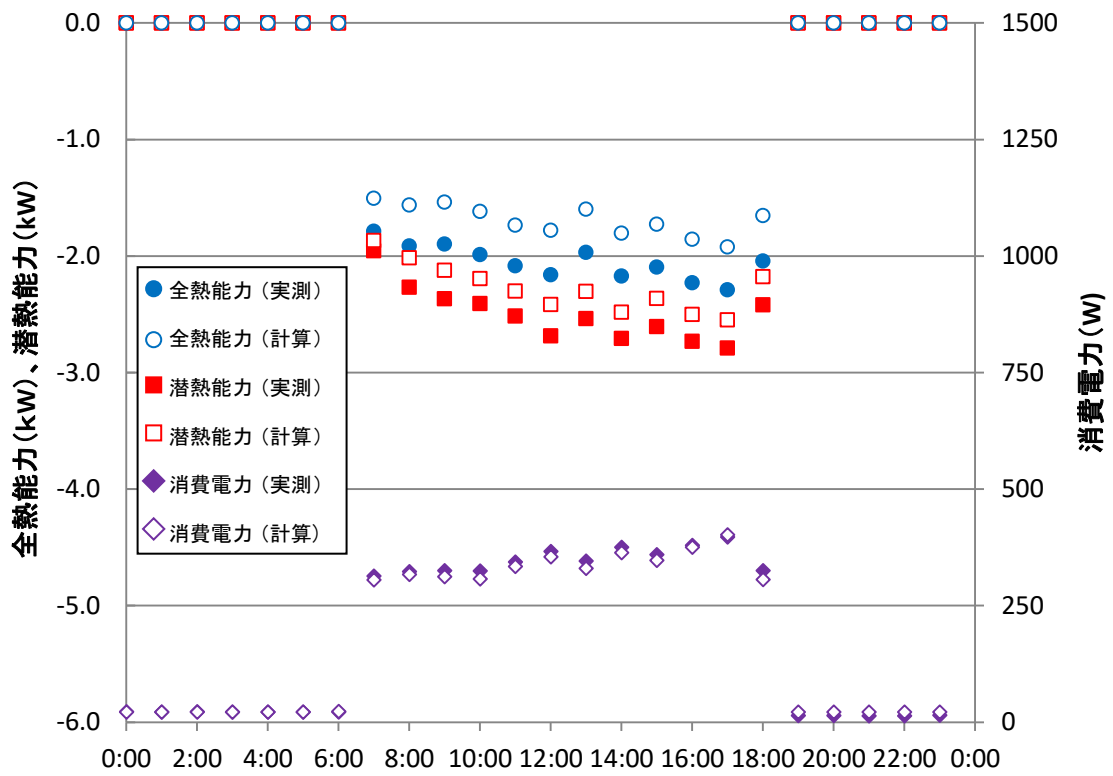


図 65 実測結果，計算結果比較 (9/5)

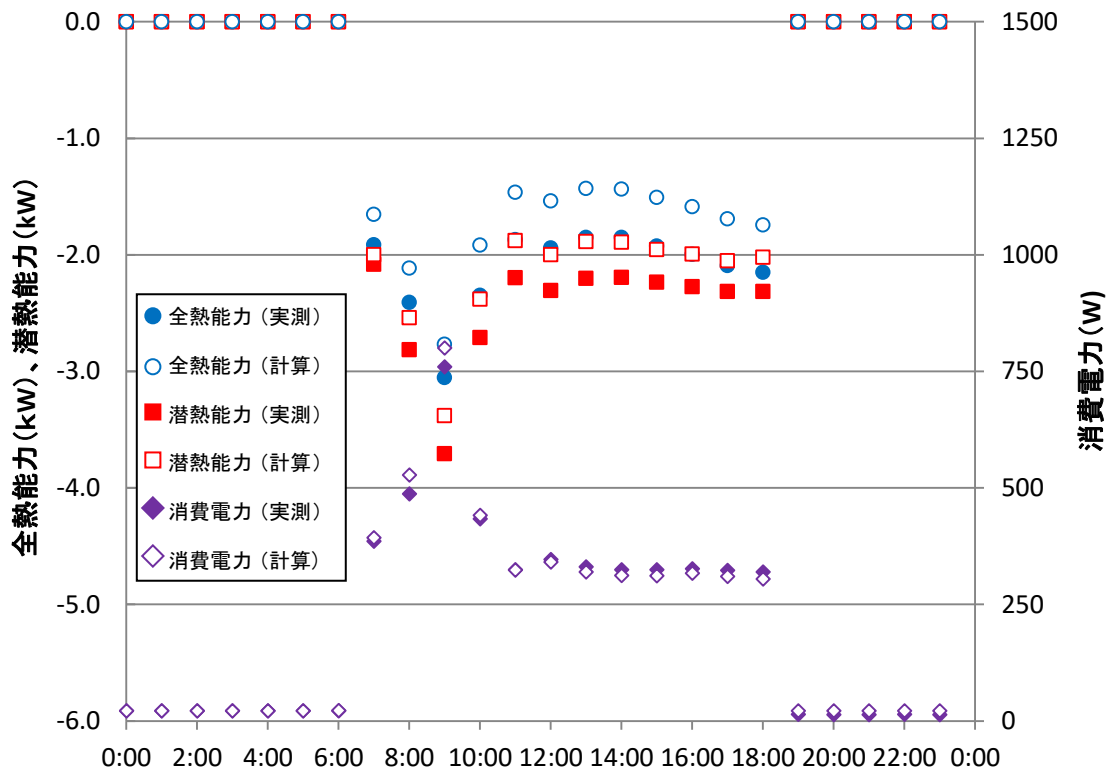


図 66 実測結果，計算結果比較 (9/8)

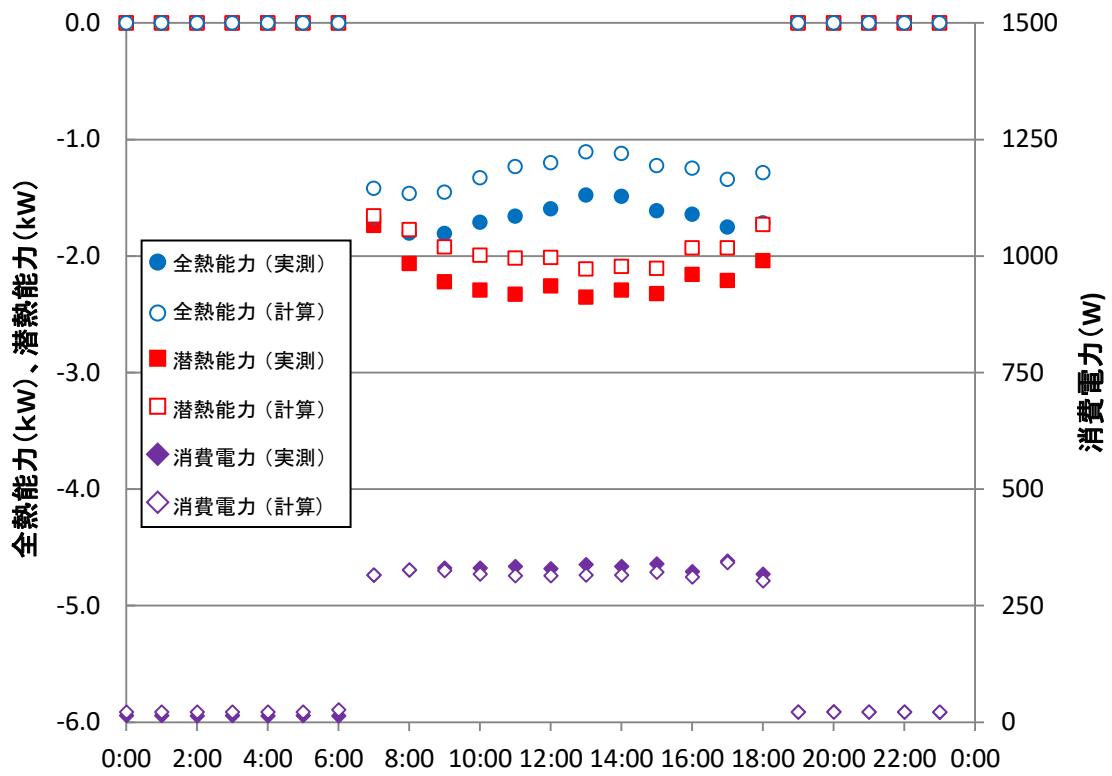


図 67 実測結果，計算結果比較 (9/9)

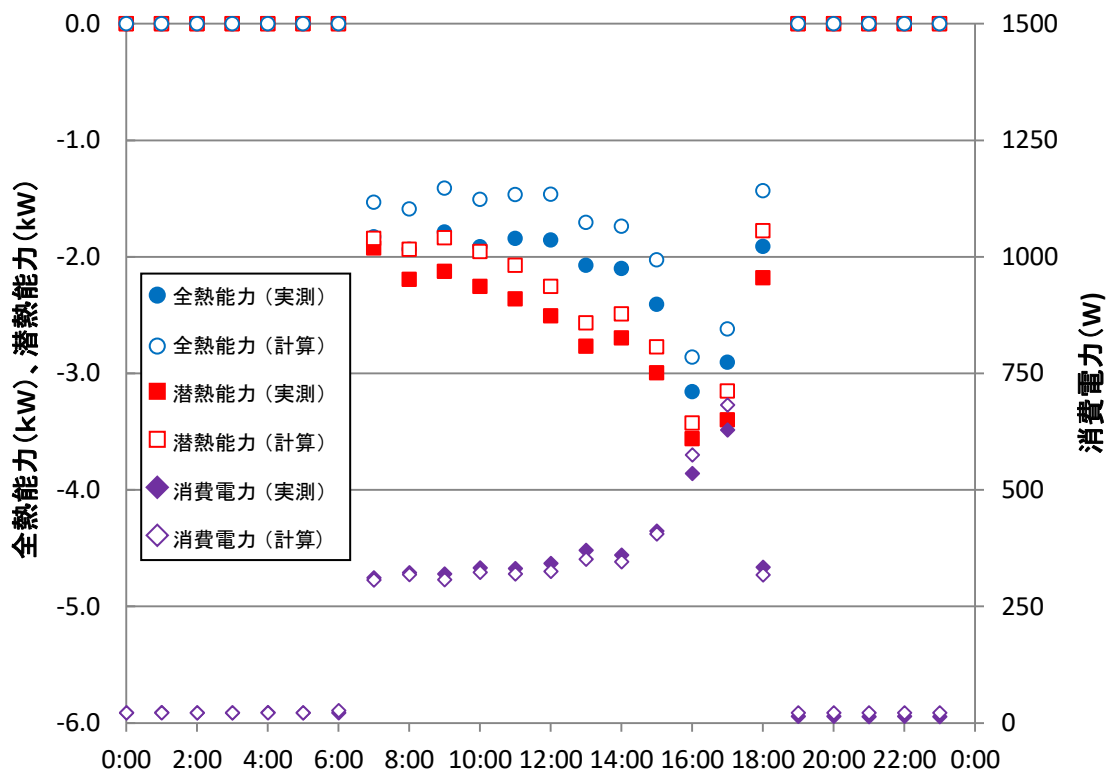


図 68 実測結果，計算結果比較 (9/10)

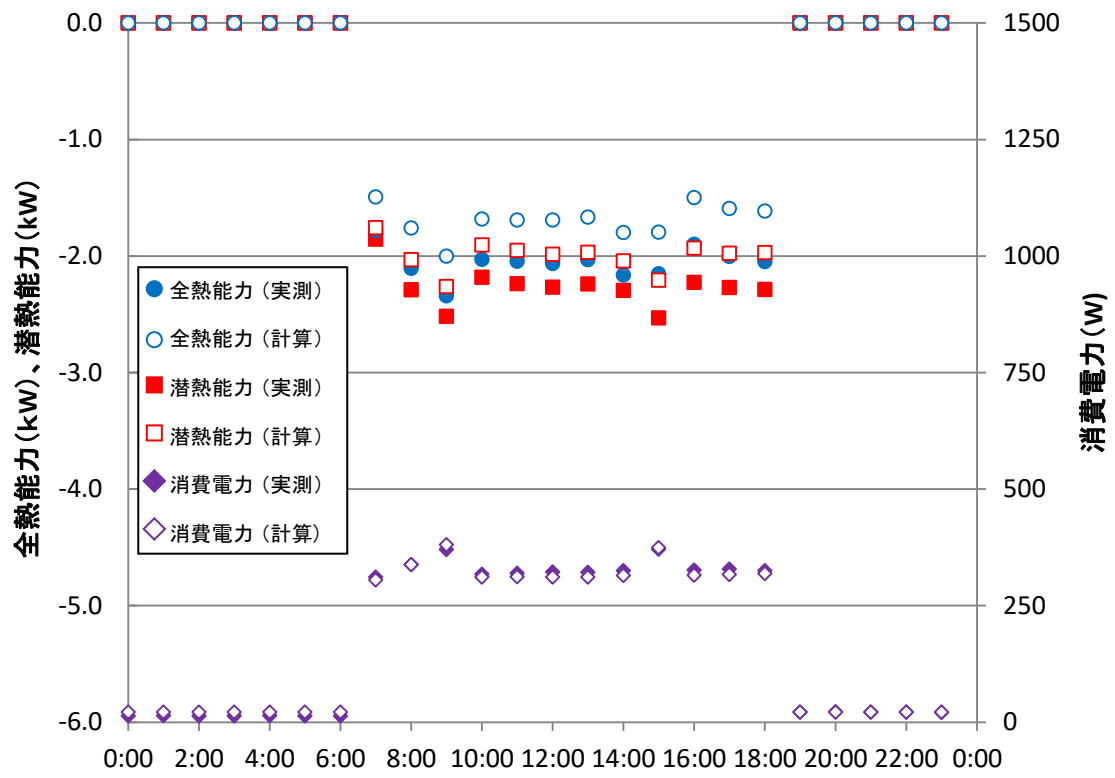


図 69 実測結果, 計算結果比較 (9/11)

付録 C

住宅における居住者の通風・冷房行為に関する実態把握と行為モデルの構築

期間実測調査による住宅の夏期室内温熱環境に関する実態データの収集

1 期間実測調査による住宅の夏期室内温熱環境に関する実態データの収集

1.1 調査方法

調査概要を表 1 に示す。本調査の実施期間は、2014 年 7 月 1 日から同年 10 月 10 日までの間とした。調査対象は、両親と同居する大学生・大学院生の自宅とし、表 1 に示す要件を満たす 10 世帯とした。対象居室は、居間と寝室とした。

表 1 調査概要

調査方法	環境データ、エアコン・扇風機・窓のステータスデータ、在室状況の自動記録
対象世帯	調査対象の要件 <ul style="list-style-type: none"> ・室内飼いペットがいない ・家族に自宅療養者や要介護者がいない ・居間と寝室の両方にエアコンが設置されている ・居間と寝室が分かれている(同じ部屋でない) ・夏の間、居間を寝室として使うことが少ない
対象居室	居間、寝室
調査期間	2014/7/1~2014/10/10

調査項目は、外気温度、室内温度、室内湿度、エアコンの稼働状況、扇風機の稼働状況、窓の開閉状況、在室状況とした。このうち、外気温度、室内温度、室内湿度、エアコンの稼働状況、窓の開閉状況、在室状況については各種計測機器により自動でデータを収集した。計測概要を表 2 に、設置例を図 1 に示す。外気温度の計測は、ロガー付き小型温度計 (Onset U23-002) を用いて軒下もしくはベランダで行った。室内温度および室内湿度の計測は、ロガー付き小型温湿度計 (Onset UX100-011) により行った。エアコンの稼働状況は、エアコン吹き出し口にロガー付き小型温湿度計 (Onset UX120-014M) を設置して吹出し空気温度を計測し、後の解析においてその変化により判断した。扇風機の稼働状況は、ロガー付き磁気センサ (Onset UX90-004M) により記録した。窓の開閉状況は、ロガー付きマグネットセンサ (Onset UX90-001M) により記録した。在室状況については、室内 CO₂ 濃度を NDIR 方式 CO₂ センサ (T&D RTR-576-H) により計測し、後の解析においてその変化により判断した。計測間隔は外気温度および在室状況については 10 分とし、その他の項目については 1 分とした。なお、機器の設置は居住者自身で行うものとし、外気温度、室内温度および室内湿度については、直射日光やエアコン室外機・室内機およびその他家電機器からの影響ができるだけ小さい場所に、CO₂ センサについては滞在場所付近に設置するように依頼した。

表 2 計測概要

計測項目	機器名称	センサー種別	計測箇所数	記録間隔
外気温度	Onset U23-002	バンドギャップ式	各世帯1点	10分
室内温度	Onset UX100-011	バンドギャップ式	各室1点	1分
室内湿度		静電容量式		1分
エアコンの稼働状況	Onset UX120-014M	熱電対	エアコン1台につき1点	1分
扇風機の稼働状況	Onset UX90-004M	磁気センサ	扇風機1台につき1点	1分
窓の開閉状況	Onset UX90-001M	マグネットセンサ	開閉箇所1つにつき1点	1分
在室状況	T&D RTR-576-H	NDIR方式CO2センサ	各室1点	10分

さらに、調査対象の属性と居住環境や周辺環境に関するアンケートを行った。設問項目を表 3 に示す。なお、間取りについては、以下の情報を含むように依頼し、機器の設置場所と発熱源、エアコンや窓との位置関係を把握した。

- 部屋の大体の大きさ（縦・横のサイズ）
- 大型の電化製品（テレビ、冷蔵庫、エアコン等）
- 家具（テーブル、ソファ、ベッド等）の位置
- 窓・勝手口・室内扉の位置とサイズ
- 窓については種類（以下の窓種類の選択肢から選択）と防犯用面格子の有無
- 外気温湿度、室内温湿度、CO₂センサについては、センサの設置場所と床上からの高さ
開閉センサについては、室内側からみた大よその設置位置と設置したセンサの ID

間取りに関する回答の記入例を図 1 に示す。

表 3 設問項目の一覧

質問項目		回答方法	入力値/選択肢
世帯属性	性別	選択	男性、女性
	年齢	入力	数値
	職業	選択	勤め人、自営業者、農林水産業者、アルバイト・パート、専業主婦・主夫、小学生、中学生、高校生、大学生、その他
住宅建物	建築年	入力	数値
	住宅形式	選択	戸建、集合
	総階数	入力	数値
	居間/主寝室のある階	入力	数値
	住戸の位置	選択	中住戸、角住戸
	間取り	入力	図示
住居環境に対する印象	風通し	選択	良い、やや良い、どちらとも言えない、やや悪い、悪い
	日当たり	選択	
周辺環境に対する印象	騒音	選択	気にならない、あまり気にならない、どちらとも言えない、やや気になる、非常に気になる
	大気汚染、砂ぼこり	選択	
	臭気	選択	
	人通り(日中・夜間)	選択	少ない、やや少ない、どちらとも言えない、やや多い、多い
	車通り(日中・夜間)	選択	
真夏におけるエアコンオフタイマーの利用状況	治安	選択	良い、やや良い、どちらとも言えない、やや悪い、悪い
	利用頻度	選択	毎日、週に4~6日、週に1~3日、週に0回
生活スケジュール	オフタイマーの設定時間	選択	数値
	在宅時間・就寝時間	入力	図示

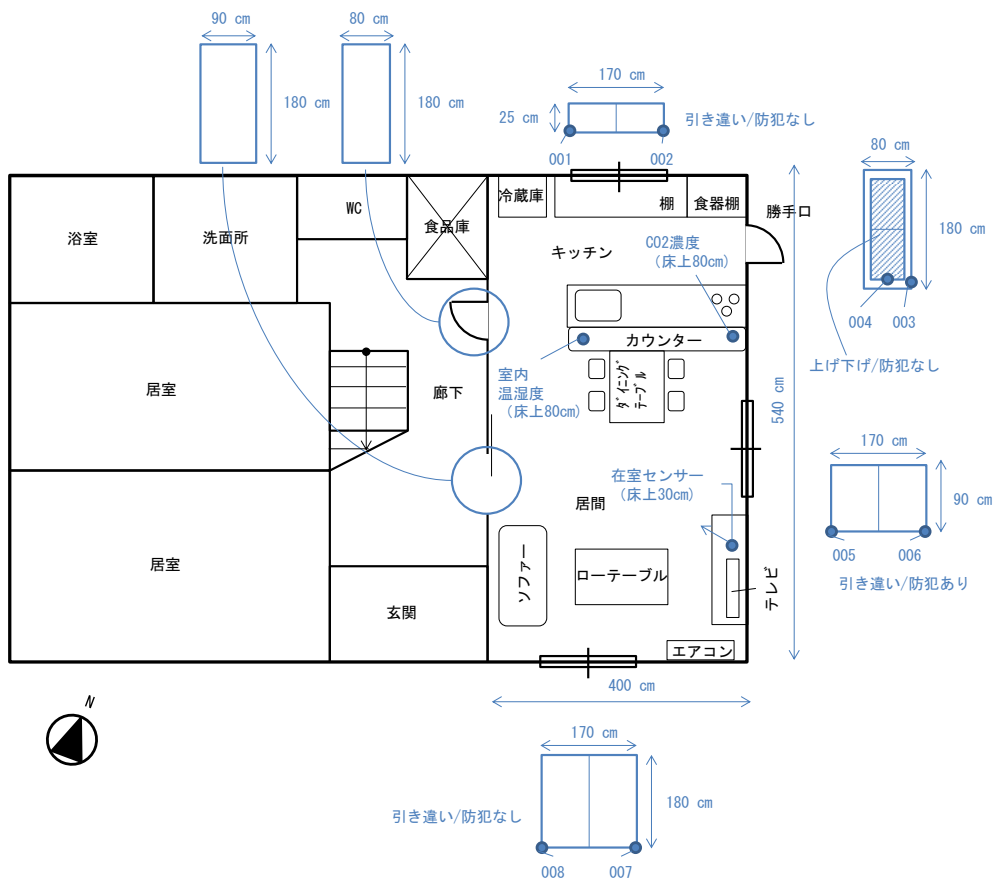


図 1 間取りに関する回答の記入例

1.2 調査対象の基本属性

調査対象世帯の家族構成を表 4 に、入居中の住宅建物に関する状況を表 5 に、住居環境に関する印象を表 6 に、周辺環境に関する印象を表 7 に、真夏におけるエアコンオフタイマーの利用状況を表 8 に示す。

調査世帯 10 世帯のうち、世帯 ID : 4、世帯 ID : 6、世帯 ID : 8 の計 3 世帯は妻が専業主婦、その他の 7 世帯は共働であった。また、世帯 ID : 5 は第 1 子と第 2 子の年齢が他の世帯に比べて高く、世帯 4 は高齢者が含まれていた。入居中の住宅建物については、7 世帯が戸建住宅であった。また、集合住宅に住まう世帯については、世帯 ID : 1 を除けば 1~2 階に住戸が位置していた。住居環境については、世帯 ID : 2 と世帯 ID : 3 は寝室の、世帯 ID : 5 は居間と寝室の風通しに対して悪い側の印象を有していた。周辺環境については、世帯 ID : 5 は騒音に対して、世帯 ID : 8 は騒音・大気汚染・臭気に対して「やや気になる」と回答した。治安については、いずれの世帯の回答も「良い」「やや良い」であった。人通りや車通りについては、日中の車通りに対して世帯 ID : 1、世帯 ID : 6 および世帯 ID : 9 が多い側の回答をしたが、全体としては少ないという印象を有していた。エアコンオフタイマーの利用については、半数の世帯（世帯 ID : 2、世帯 ID : 4、世帯 ID : 5、世帯 ID : 7、世帯 ID : 10）が利用していた。

表 4 調査対象世帯の家族構成

世帯ID	夫		妻		第1子		第2子		第3子		その他	
	職業	年齢	職業	年齢	職業	年齢	職業	年齢	職業	年齢	職業	年齢
1	勤め人	46	勤め人	45	大学生	19						
2	勤め人	47	アルバイト(パート)	46	大学生	21	中学生	15				
3	勤め人	55	アルバイト(パート)	52	大学生	20						
4	勤め人	51	専業主婦	47	大学生	19	高校生	17	中学生	15	専業主婦	78
5	勤め人	58	勤め人	55	アルバイト(パート)	27	勤め人	24	大学生	22		
6	勤め人	49	専業主婦	49	大学生	21	中学生	15				
7	勤め人	56	アルバイト(パート)	52	大学生	25	大学生	21				
8	勤め人	53	専業主婦	53	大学院生	23	大学生	22				
9	勤め人	52	アルバイト(パート)	49	大学院生	23	大学生	20				
10	勤め人	57	アルバイト(パート)	52	大学院生	23	大学生	20				

表 5 入居中の住宅建物

世帯ID	建築年	住宅形式	住棟の 総階数 [階]	居間の 位置 [階]	寝室の 位置 [階]	住戸位置 (集合の場合)
1	2001	集合	14	14	14	中間住戸
2	1999	戸建	3	2	1	-
3	1982	戸建	2	1	2	-
4	2003	戸建	2	1	2	-
5	不明	戸建	3	2	2	-
6	1986	集合	3	1	2	角住戸
7	1998	戸建	2	1	2	-
8	1977	戸建	2	1	2	-
9	1996	集合	5	2	2	角住戸
10	2003	戸建	2	2	2	-

表 6 住居環境に関する印象

世帯ID	居間		寝室	
	風通し	日当たり	風通し	日当たり
1	良い	良い	やや良い	悪い
2	どちらとも言えない	どちらとも言えない	やや悪い	悪い
3	どちらとも言えない	やや悪い	やや悪い	良い
4	良い	良い	やや良い	やや悪い
5	やや悪い	どちらとも言えない	悪い	悪い
6	やや良い	やや悪い	やや良い	やや良い
7	やや良い	良い	良い	良い
8	やや良い	やや良い	良い	良い
9	やや良い	やや良い	やや良い	やや悪い
10	どちらとも言えない	良い	やや良い	良い

表 7 周辺環境に関する印象

世帯ID	騒音	大気汚染	臭気	治安
1	あまり気にならない	気にならない	あまり気にならない	やや良い
2	気にならない	あまり気にならない	気にならない	やや良い
3	気にならない	気にならない	気にならない	良い
4	気にならない	気にならない	気にならない	どちらとも言えない
5	やや気になる	あまり気にならない	気にならない	やや良い
6	どちらとも言えない	あまり気にならない	あまり気にならない	やや良い
7	気にならない	あまり気にならない	あまり気にならない	やや良い
8	やや気になる	やや気になる	やや気になる	やや良い
9	あまり気にならない	どちらとも言えない	気にならない	やや良い
10	気にならない	気にならない	気にならない	良い

世帯ID	日中の人通り	夜間の人通り	日中の車通り	夜間の車通り
1	どちらとも言えない	やや少ない	やや多い	どちらとも言えない
2	やや少ない	少ない	少ない	少ない
3	少ない	少ない	少ない	少ない
4	やや少ない	少ない	少ない	少ない
5	少ない	少ない	やや少ない	やや少ない
6	やや少ない	少ない	多い	やや少ない
7	やや少ない	少ない	やや少ない	少ない
8	やや少ない	やや少ない	やや少ない	少ない
9	少ない	少ない	やや多い	やや少ない
10	少ない	少ない	少ない	少ない

表 8 真夏におけるエアコンオフタイマーの利用状況

世帯ID	調査期間中における 総エアコン使用時間[h]	
	居間	寝室
1	453	476
2	264	177
3	0	0
4	172	238
5	261	0
6	752	767
7	166	33
8	475	368
9	143	0
10	642	143

1.3 分析方法

1.3.1 在室の検出

室の気積を V [m³]、換気量を Q [m³/h]、室内の CO₂ 発生量を M [m³/h]、外気の CO₂ 濃度を C_o [m³/m³]、室内の CO₂ 濃度を C [m³/m³] とすると、微小時間 dt [h] における室の CO₂ の物質収支式は次式で表される。

$$V \frac{dC}{dt} = M + Q(C_o - C)$$

式 1

式 1 を前進差分により近似すると以下の式を得る。

$$V \frac{C_{t+\Delta t} - C_t}{\Delta t} = M_t + Q_t(C_o - C_t)$$

式 2

ここで、 C_t は時刻 t における室内の CO₂ 濃度 [m³/m³]、 M_t は時刻 t における室内の CO₂ 発生量 [m³/h]、 Q_t は時刻 t における換気量 [m³/h]、 Δt は CO₂ データの時間間隔 [h] (ここでは、 $\Delta t = 10/60$ [h]) である。

在室の検出は、計測した室内の CO₂ 濃度 C_t [m³/m³] および $C_{t+\Delta t}$ [m³/m³] を用いて式 2 により推定した室内の CO₂ 発生量 M_t [m³/h] により行った。在室者 1 人当たりの CO₂ 発生量 0.02 m³/(h・人) と仮定し、室内の CO₂ 発生量の閾値を 0.02 m³/m³ とした。外気の CO₂ 濃度 C_o [m³/m³] には、計測データの最小値を用いた。換気量 Q_t [m³/h] は、室の換気回数を窓閉鎖時には 0.5 回/h、窓開放時には 2.0 回/h とし、提供された室の平面図から得られた床面積と天井高 (2.4 m を仮定) から仮定した。

1.3.2 1 分データの 10 分データへの変換

室内の温度および相対湿度については、時刻 t_i における値は、時刻 t_i から時刻 t_{i+9} までの 10 個の 1 分データを平均することで 10 分データへ変換した。エアコンの稼働状況、扇風機の稼働状況および窓の開閉状況については、1 分データにおいて時刻 t_i から時刻 t_{i+9} までの間でエアコン、扇風機、窓に対する操作が起きた場合は、10 分データではその操作の生起を時刻 t_i に割り付けた。

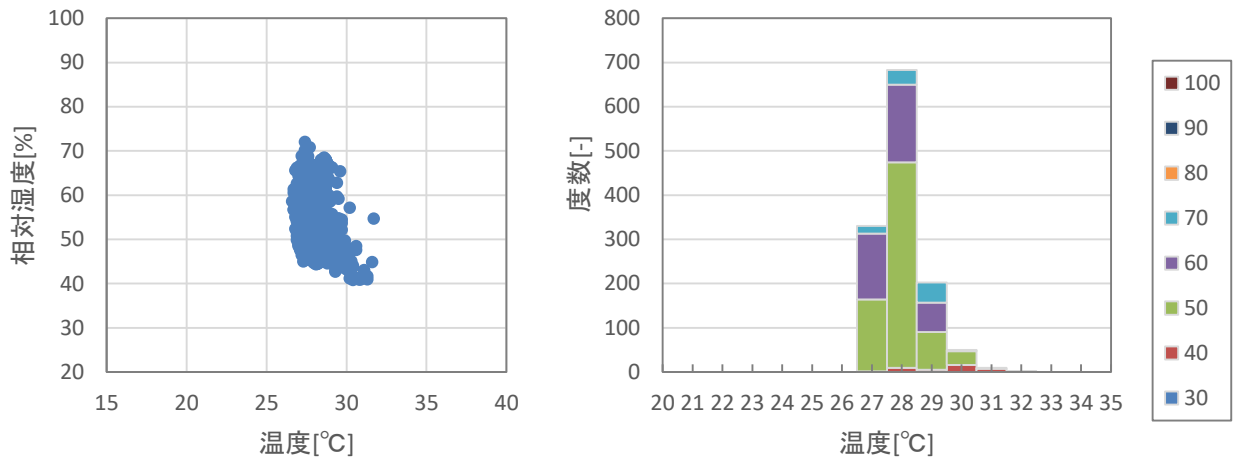
1.4 分析結果

冷房時および通風時の室内の温度および相対湿度、冷房開始時の室内温度、通風時の外気温度を世帯別・室別に図 2、図 3、図 4、図 5、図 6、図 7、図 8、図 9、図 10、図 11、図 12、図 13、図 14、図 15、図 16、図 17、図 18、図 19、図 20 および図 21 に、全世界帯について図 22 および図 23 に示す。なお、冷房時および通風時については、環境の変動が大きいことが予想される、冷房あるいは通

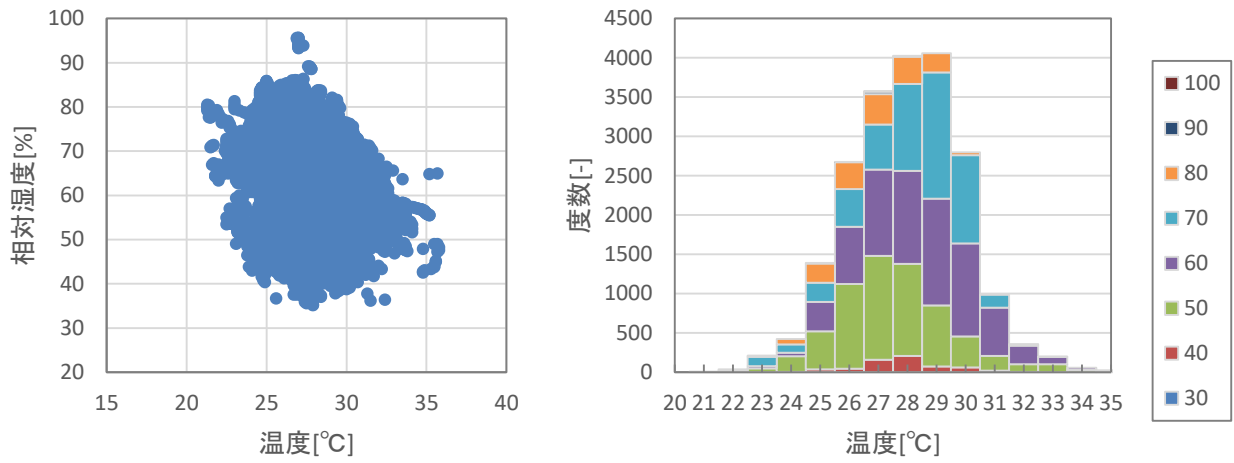
風の開始から1時間以内と、終了の30分前までのデータは除外した。

冷房時の室内温度は、 $27^{\circ}\text{C}\sim 29^{\circ}\text{C}$ の温度帯の出現頻度が高い世帯が多かった。冷房時の室内相対湿度は、60%を超える状況も発生した。冷房時に相対湿度が60%を超える状況は、特にID:6、ID:7およびD:9の居間やID:1、ID:2、ID:4、ID:6、ID:7およびID:8の寝室で他の世帯と比べて高い割合で発生した。また、居間よりも寝室で冷房時に相対湿度が60%を超える状況は発生しやすい傾向があった。冷房時の室内相対湿度について、さらに詳細に分析を行うため、室内温度が $27^{\circ}\text{C}\pm 5^{\circ}\text{C}$ の範囲にあった時の外気の温度および相対湿度を室内の相対湿度により整理した結果を図24および図25に示す。冷房時に室内温度が $27^{\circ}\text{C}\pm 5^{\circ}\text{C}$ の範囲にある場合、その大よそ3分の2において冷房時の相対湿度が60%を超える状況であった。また、冷房時の相対湿度が60%を超える状況が発生した時、外気温度は低い範囲に、外気の相対湿度は高い範囲にあることが相対的に多かった。

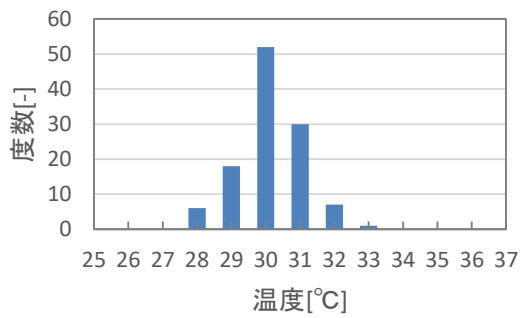
通風の使用は、ID:1、ID:2、ID:4、ID:5およびID:6の居間、ID:1、ID:4およびID:5の寝室のように室内温度が $29^{\circ}\text{C}\sim 30^{\circ}\text{C}$ を超える範囲では出現が減少した。一方で、同じ温度帯において冷房開始の出現が増加した。従って、 $29^{\circ}\text{C}\sim 30^{\circ}\text{C}$ の範囲に通風の使用と冷房の使用との分岐点が存在することが予想される。また、外気温度が 20°C を下回ると通風の使用が減少することから、この温度帯の前後に通風使用の限界となる外気温度が存在する可能性がある。



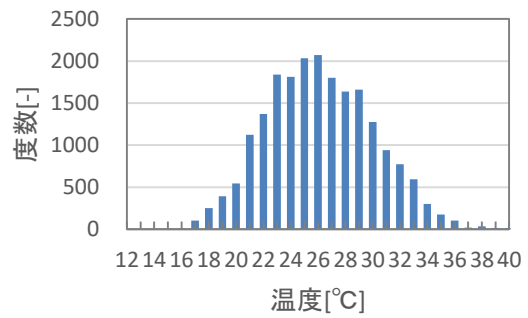
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

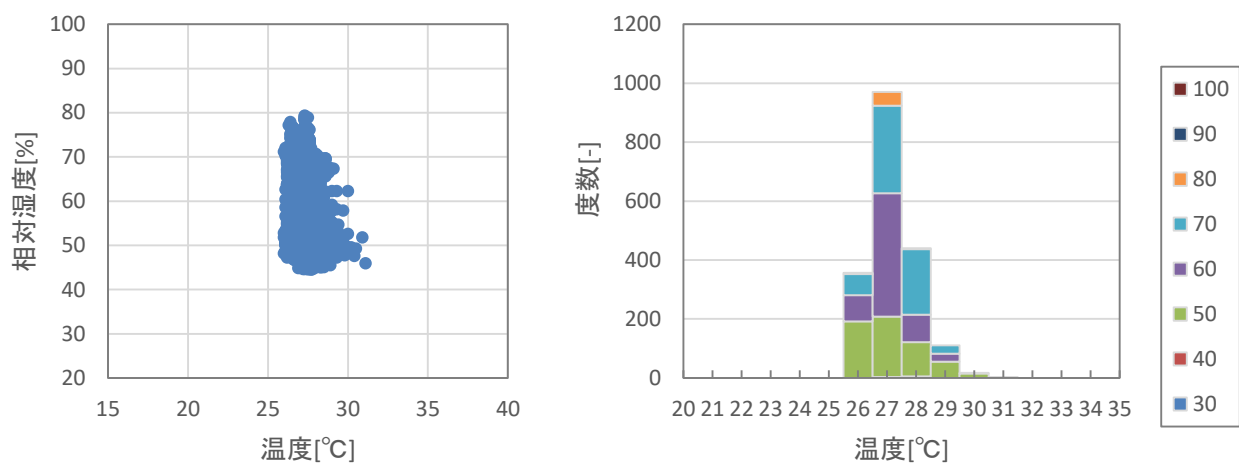


(c) 冷房開始時の室内温度

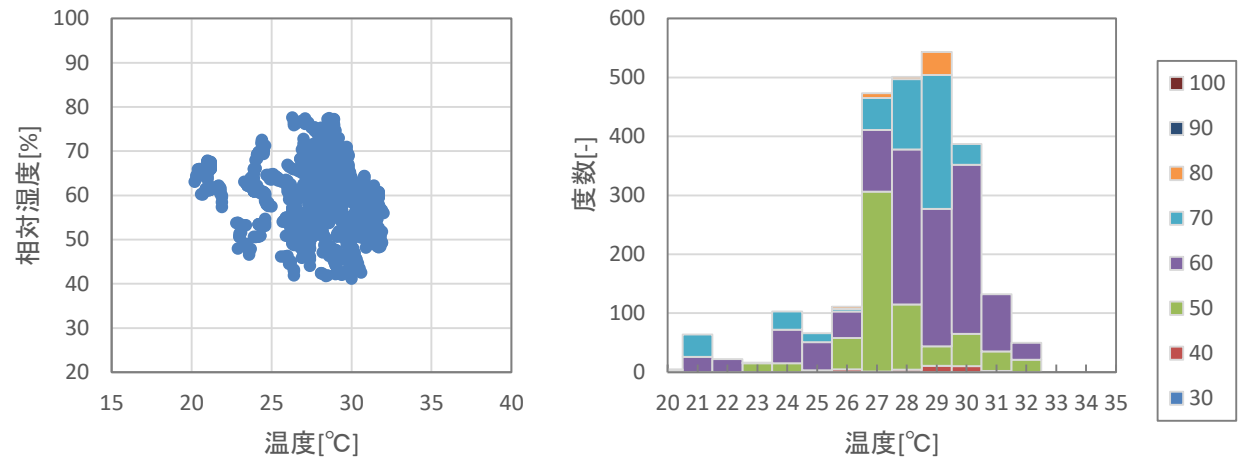


(d) 通風時の外気温度

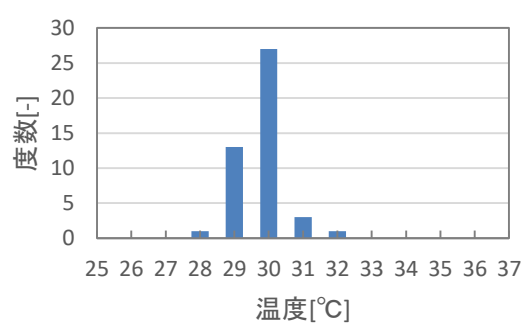
図 2 世帯 ID : 1 の居間



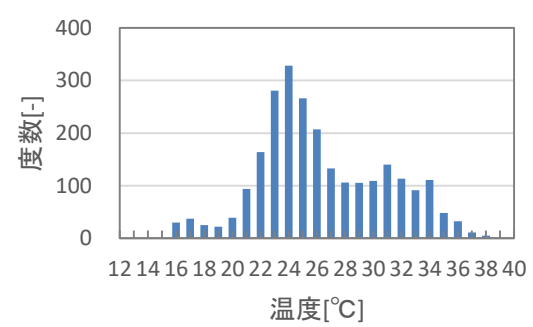
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

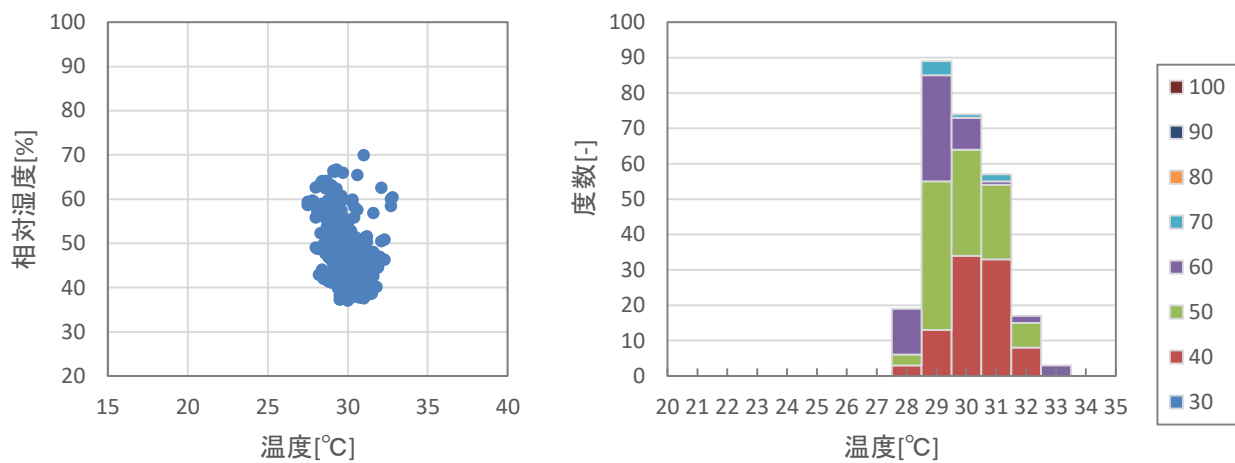


(c) 冷房開始時の室内温度

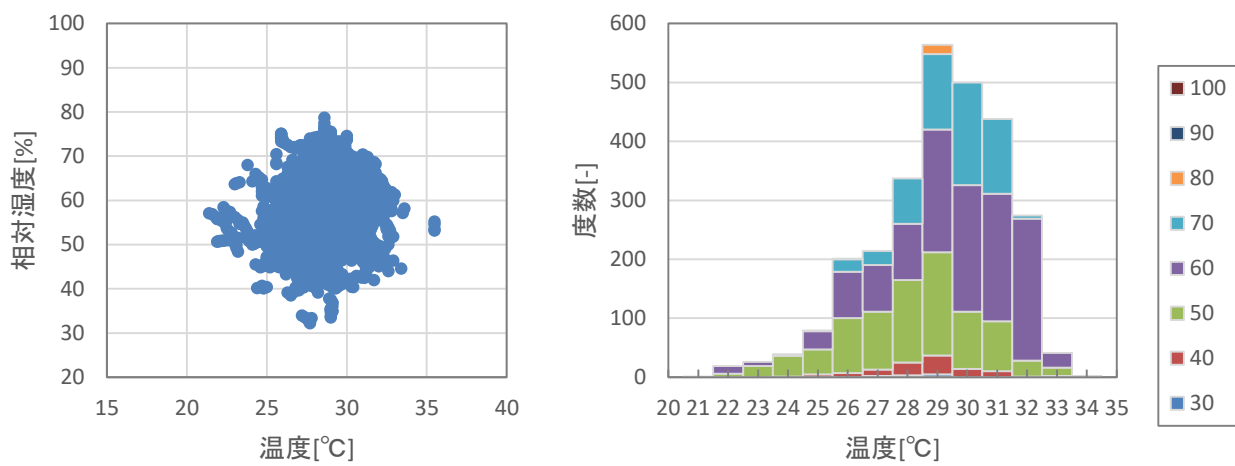


(d) 通風時の外気温度

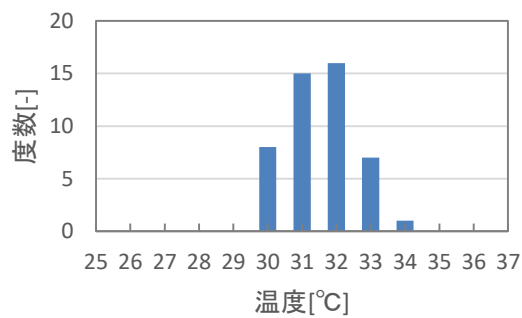
図 3 世帯 ID : 1 の寝室



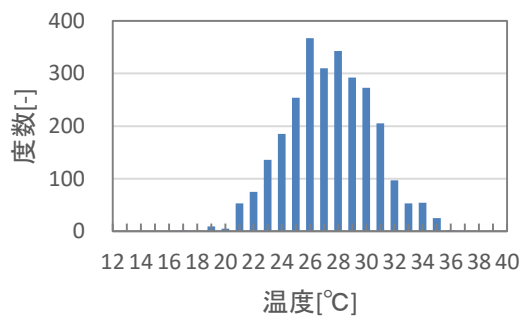
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

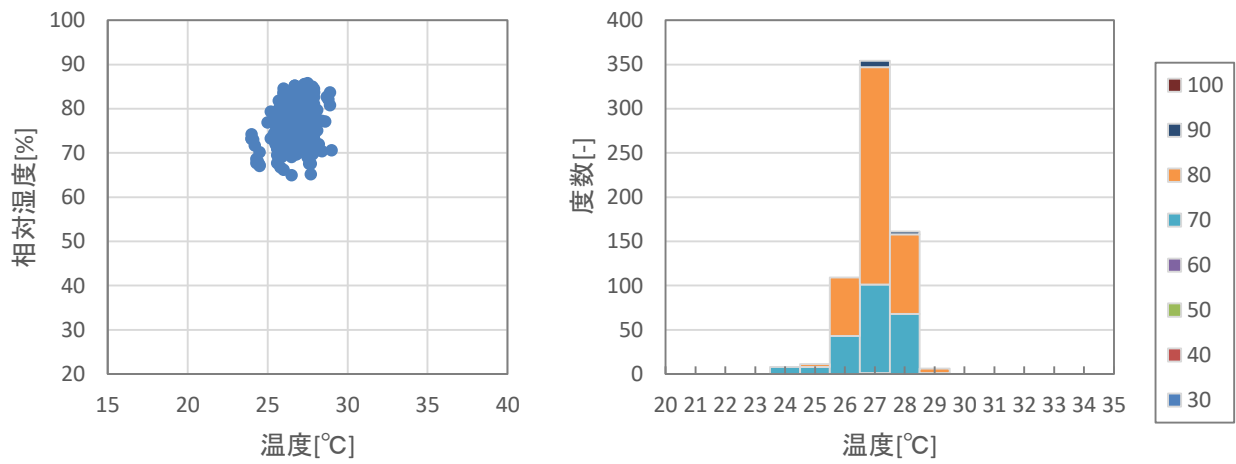


(c) 冷房開始時の室内温度

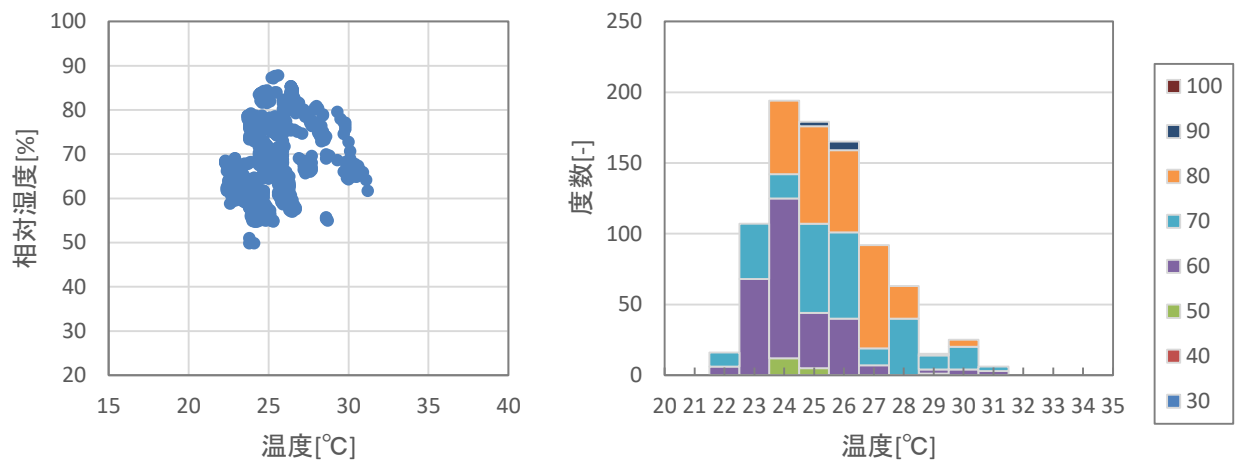


(d) 通風時の外気温度

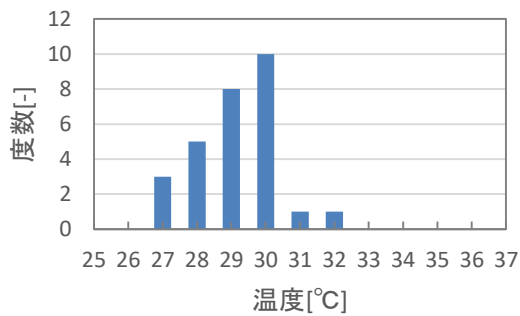
図 4 世帯 ID : 2 の居間



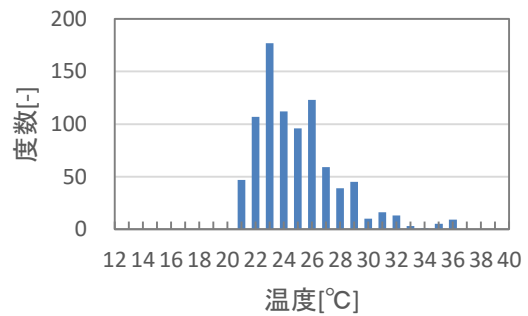
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度



(c) 冷房開始時の室内温度

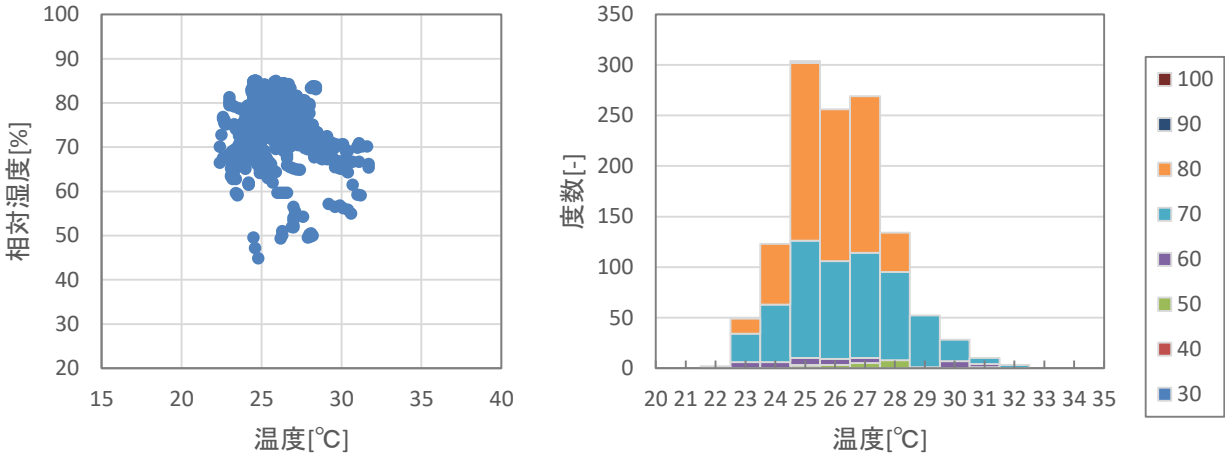


(d) 通風時の外気温度

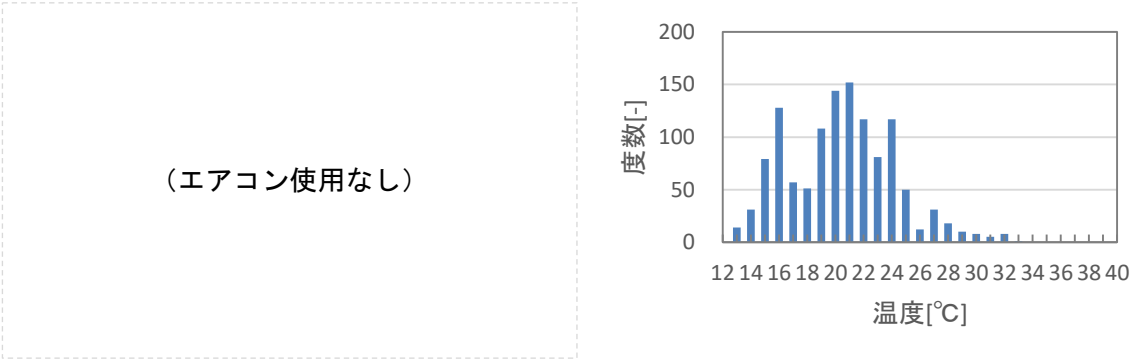
図 5 世帯 ID : 2 の寝室

(エアコン使用なし)

(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度



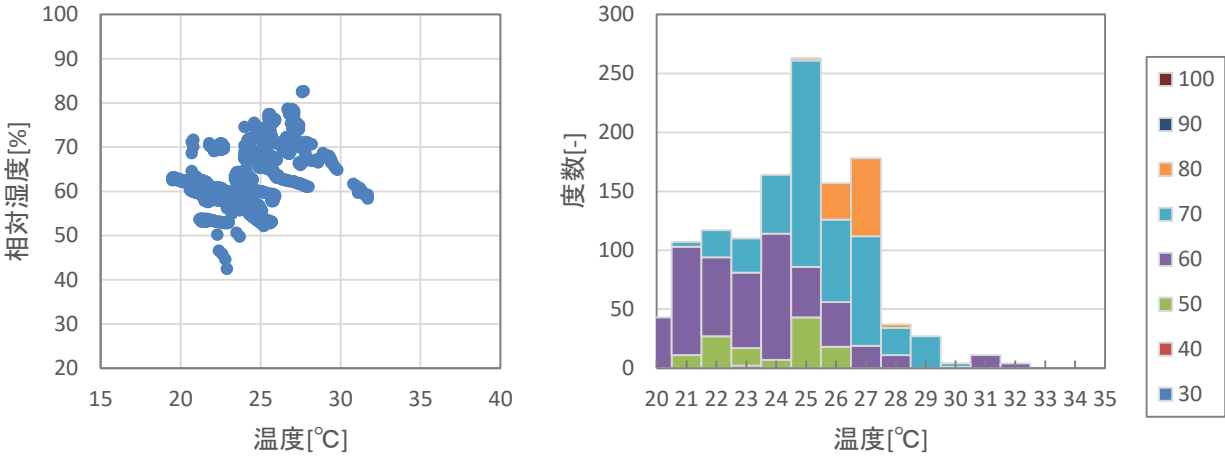
(c) 冷房開始時の室内温度

(d) 通風時の外気温度

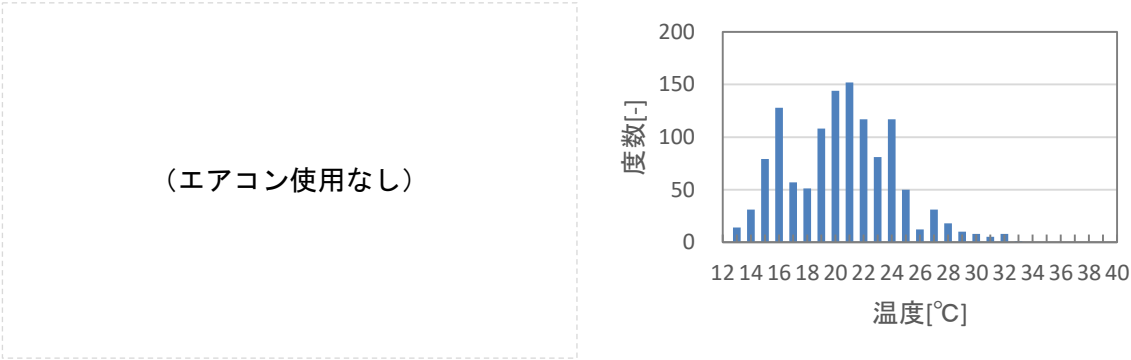
図 6 世帯 ID : 3 の居間

(エアコン使用なし)

(a) 冷房時の室内温度と相対湿度



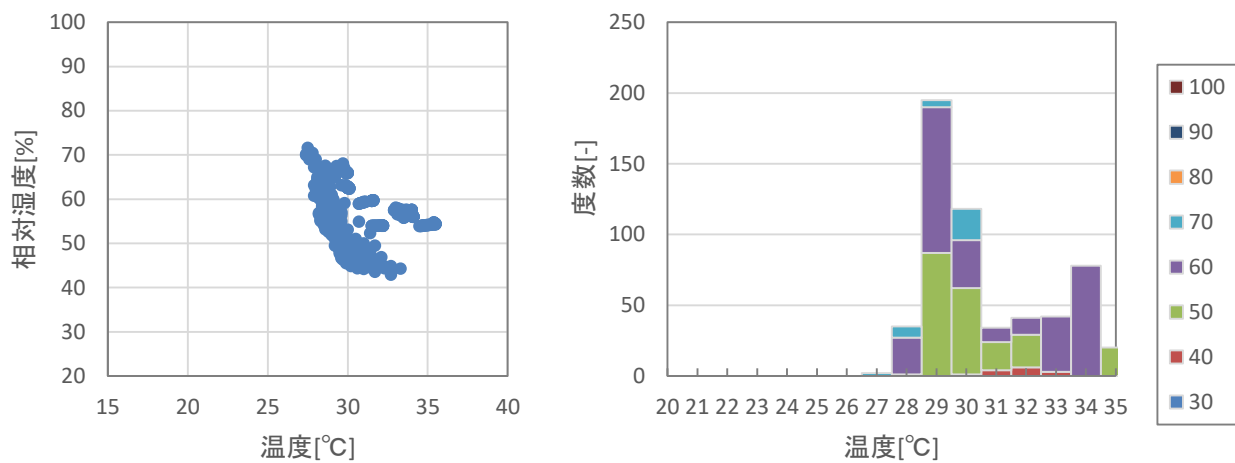
(b) 通風時の室内温度と相対湿度



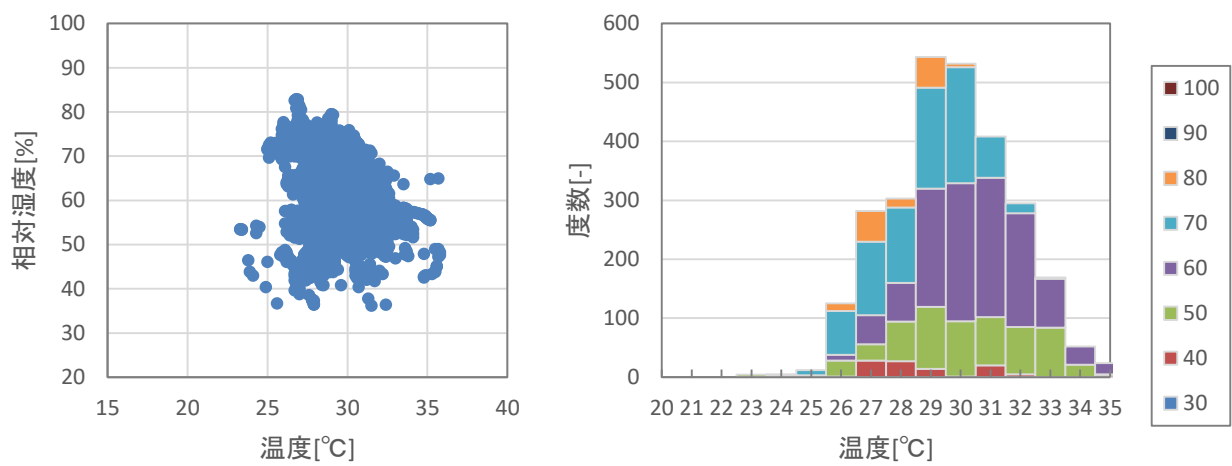
(c) 冷房開始時の室内温度

(d) 通風時の外気温度

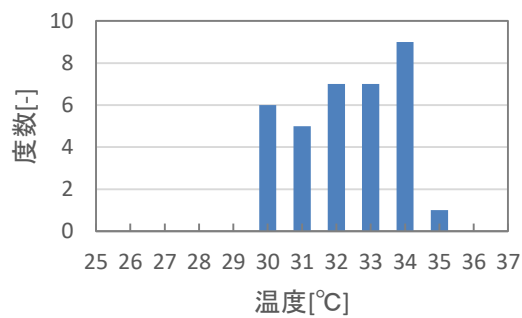
図 7 世帯 ID : 3 の寝室



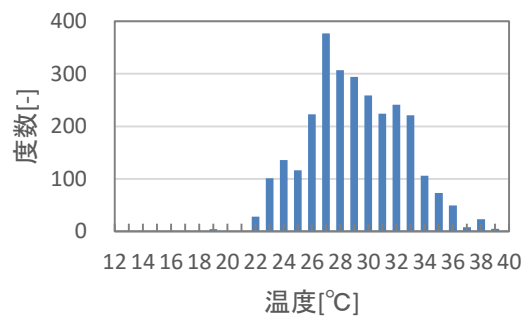
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

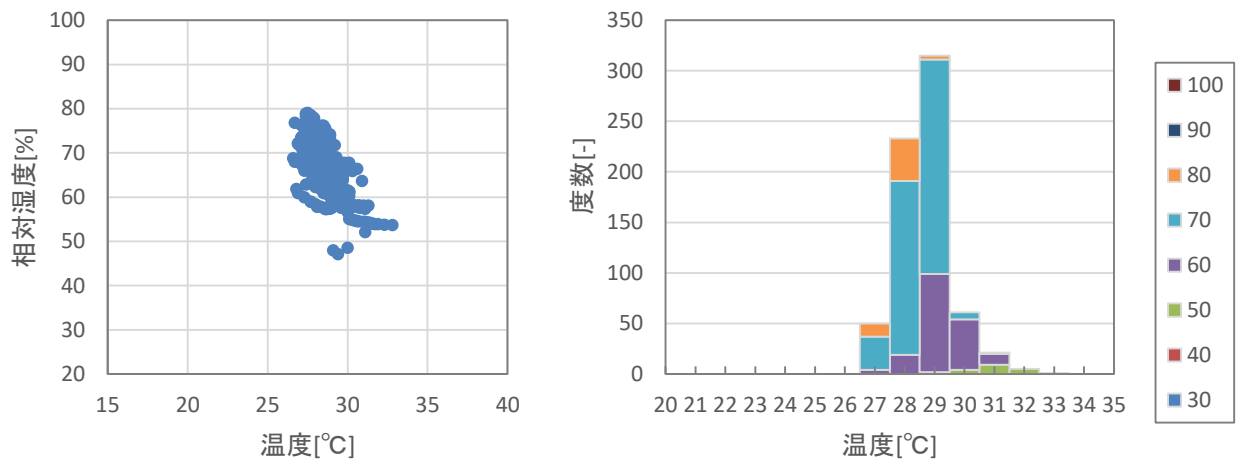


(c) 冷房開始時の室内温度

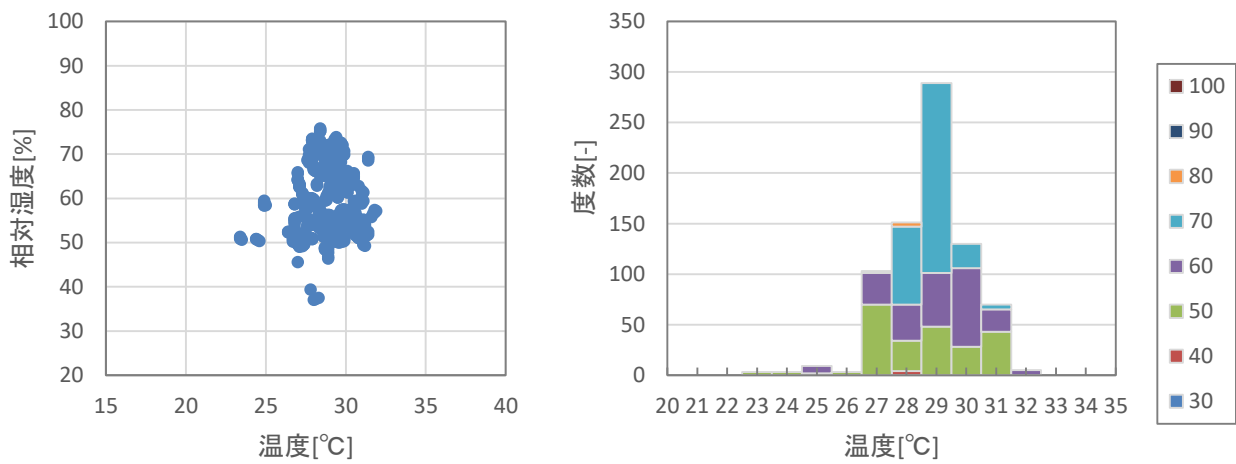


(d) 通風時の外気温度

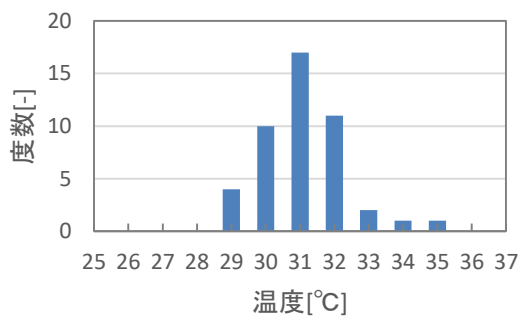
図 8 世帯 ID : 4 の居間



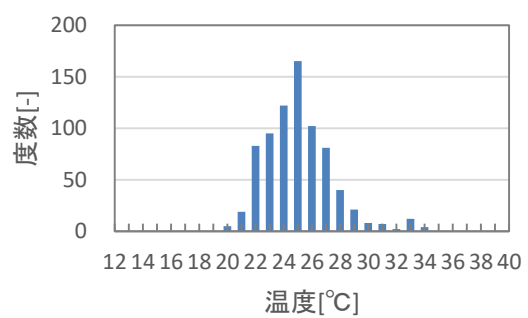
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

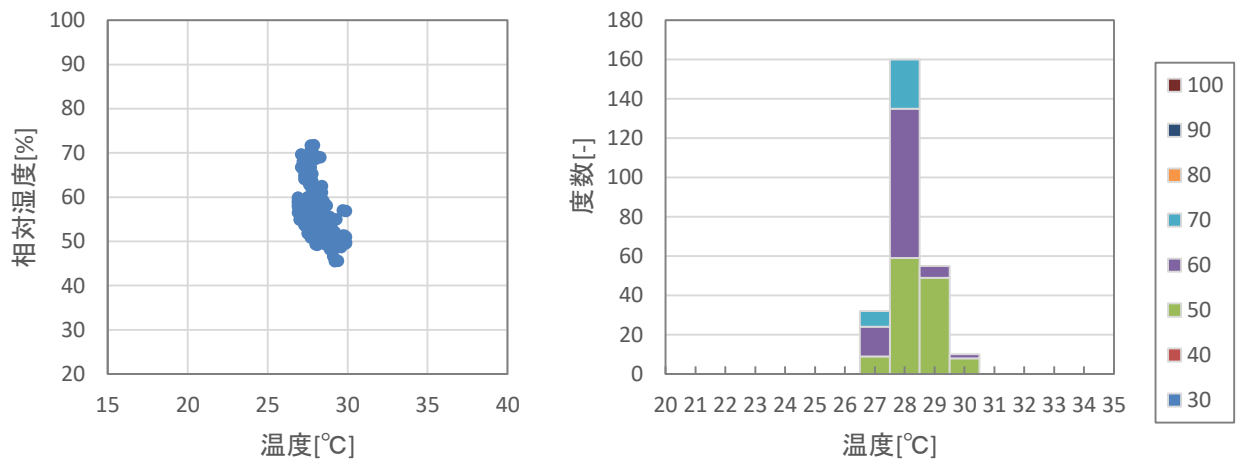


(c) 冷房開始時の室内温度

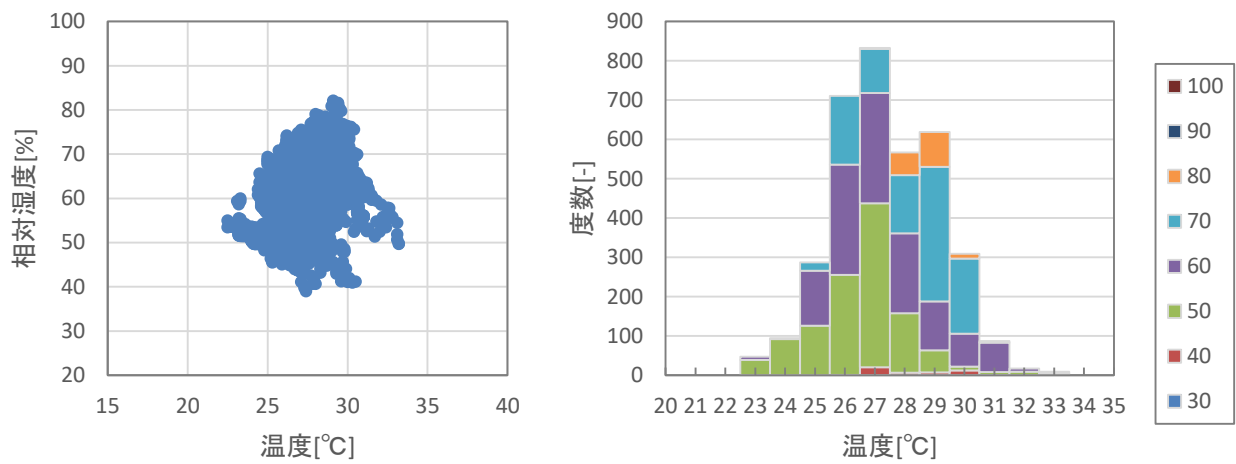


(d) 通風時の外気温度

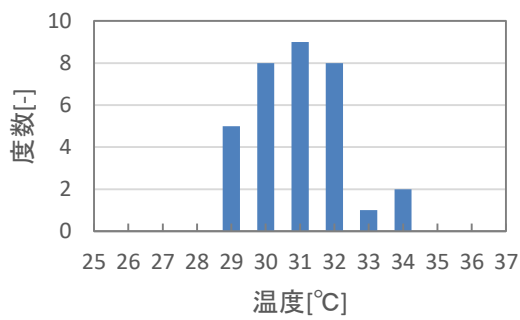
図 9 世帯 ID : 4 の寝室



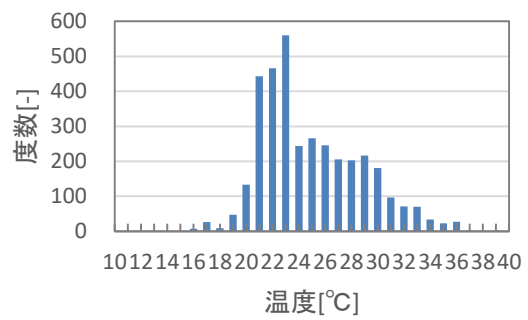
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度



(c) 冷房開始時の室内温度

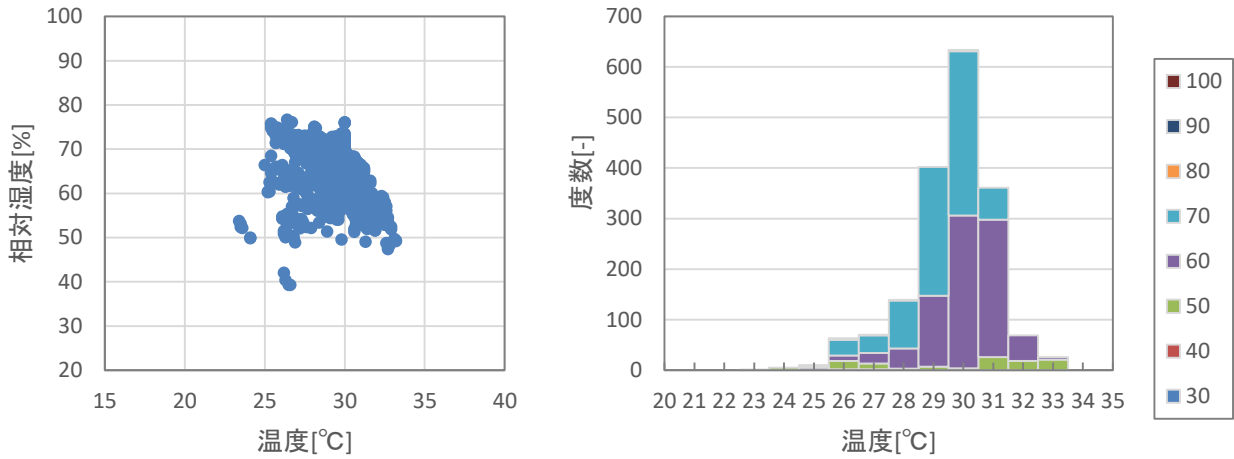


(d) 通風時の外気温度

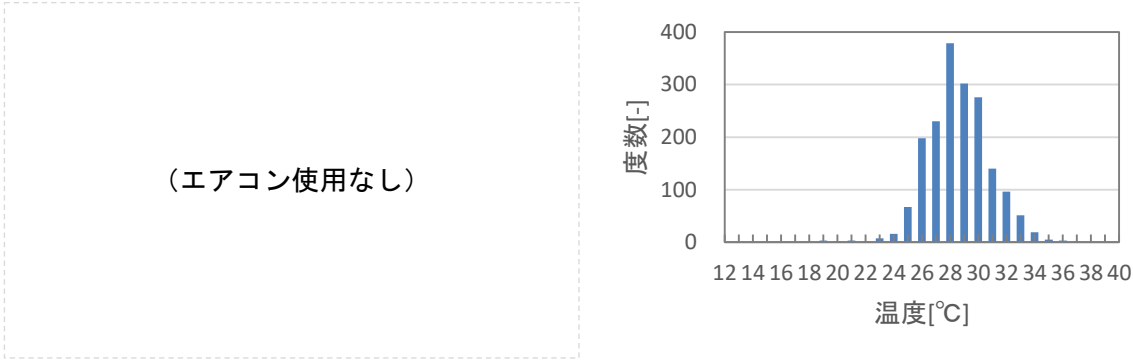
図 10 世帯 ID : 5 の居間

(エアコン使用なし)

(a) 冷房時の室内温度と相対湿度



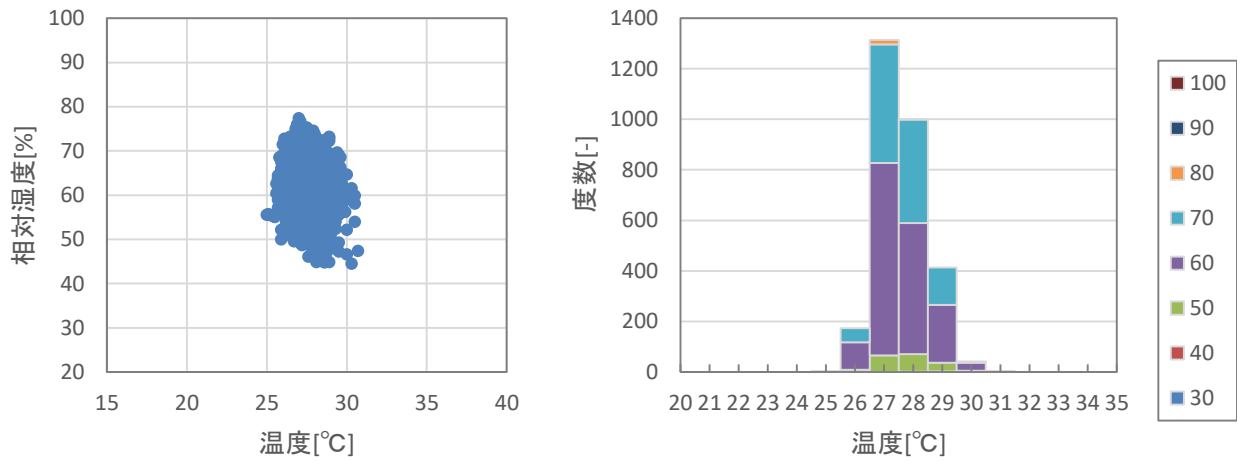
(b) 通風時の室内温度と相対湿度



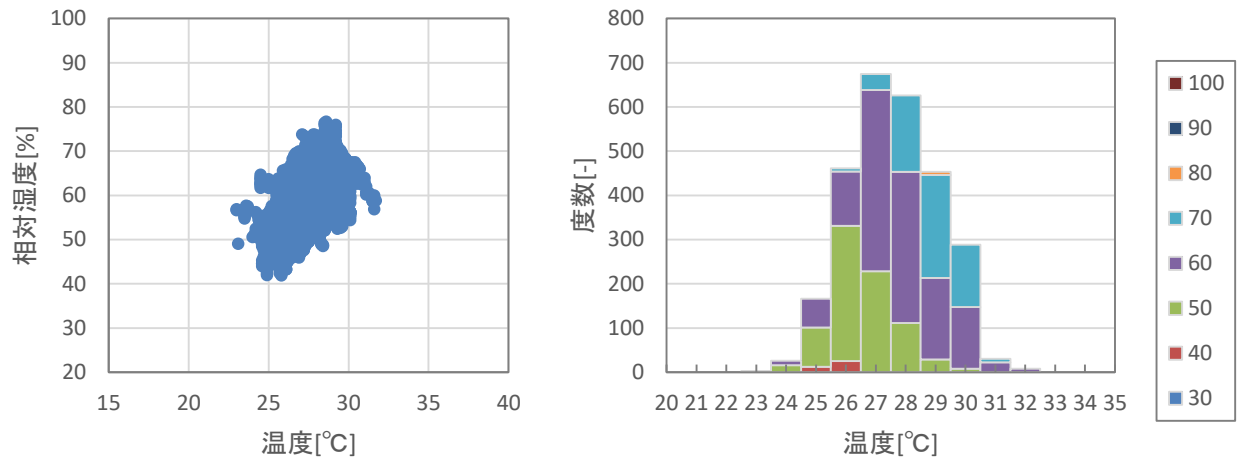
(c) 冷房開始時の室内温度

(d) 通風時の外気温度

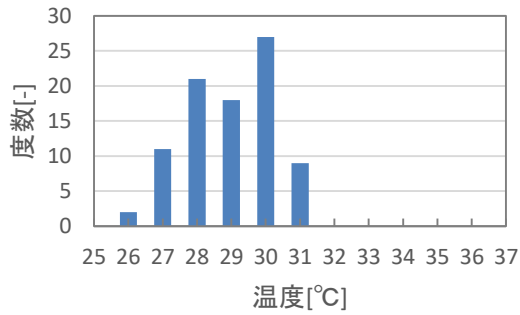
図 11 世帯 ID : 5 の寝室



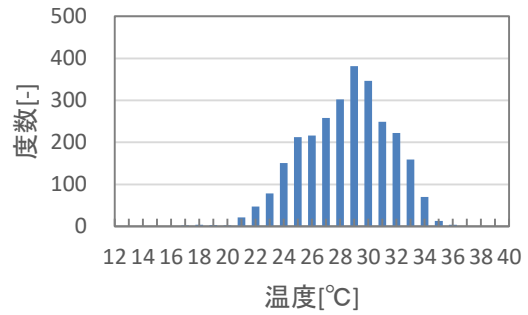
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

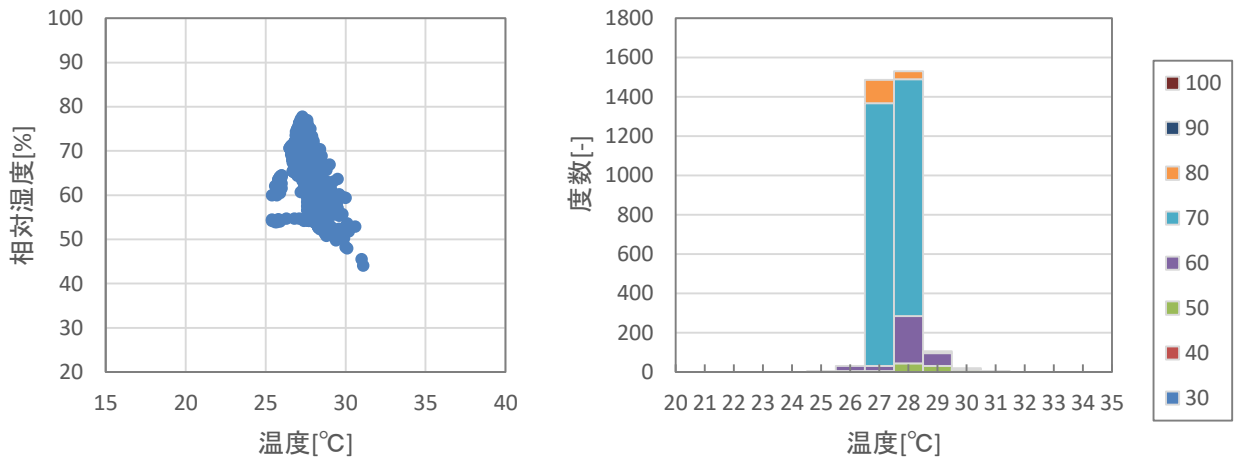


(c) 冷房開始時の室内温度

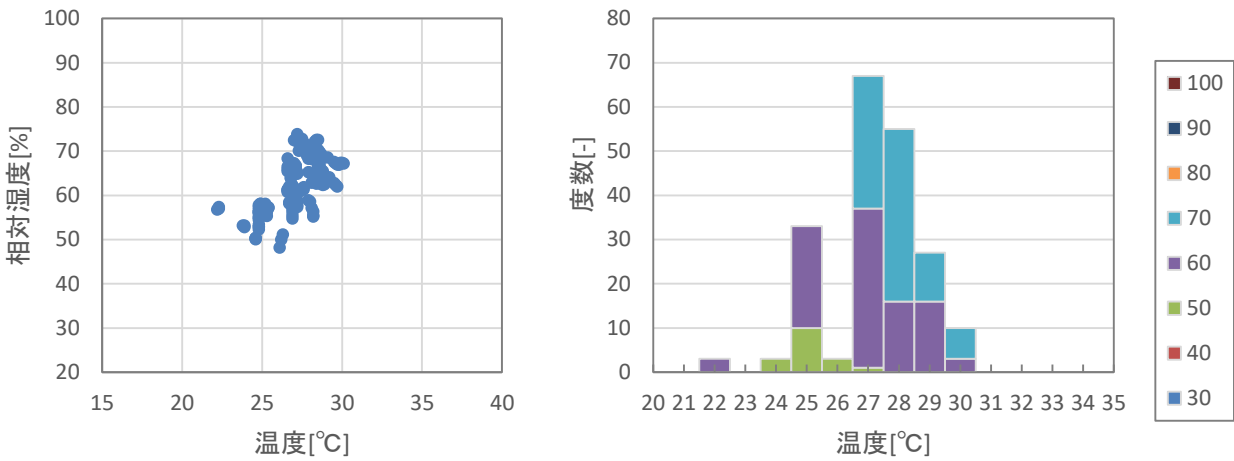


(d) 通風時の外気温度

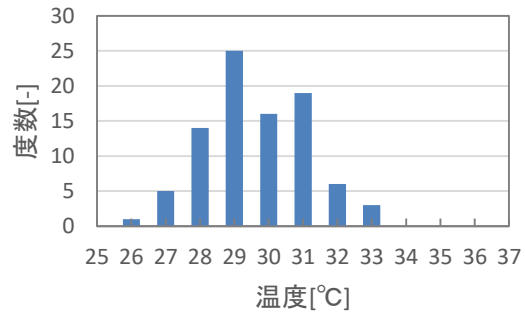
図 12 世帯 ID : 6 の居間



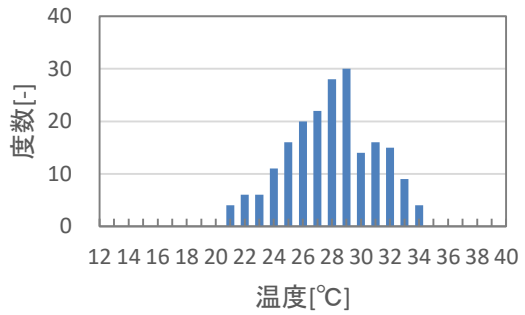
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

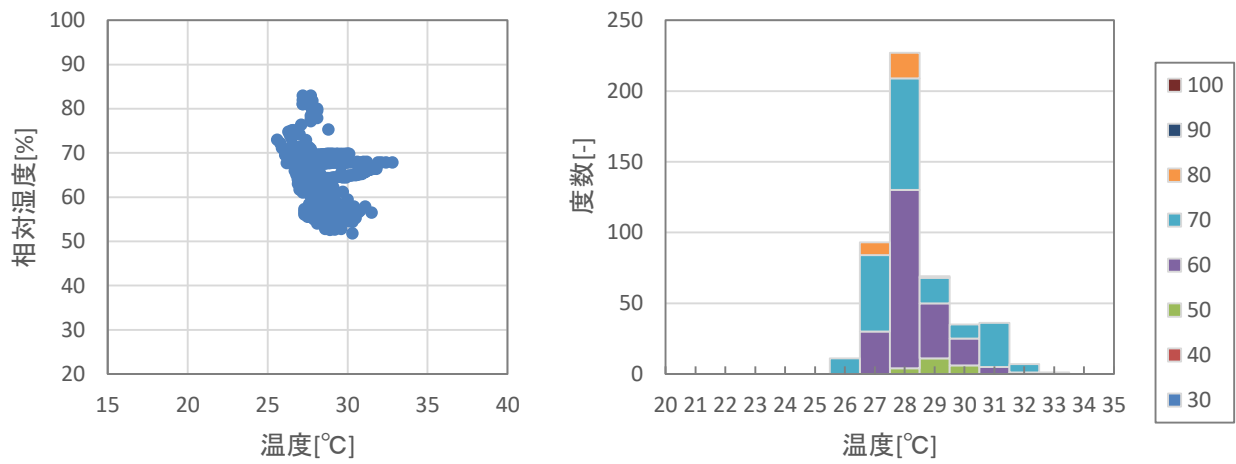


(c) 冷房開始時の室内温度

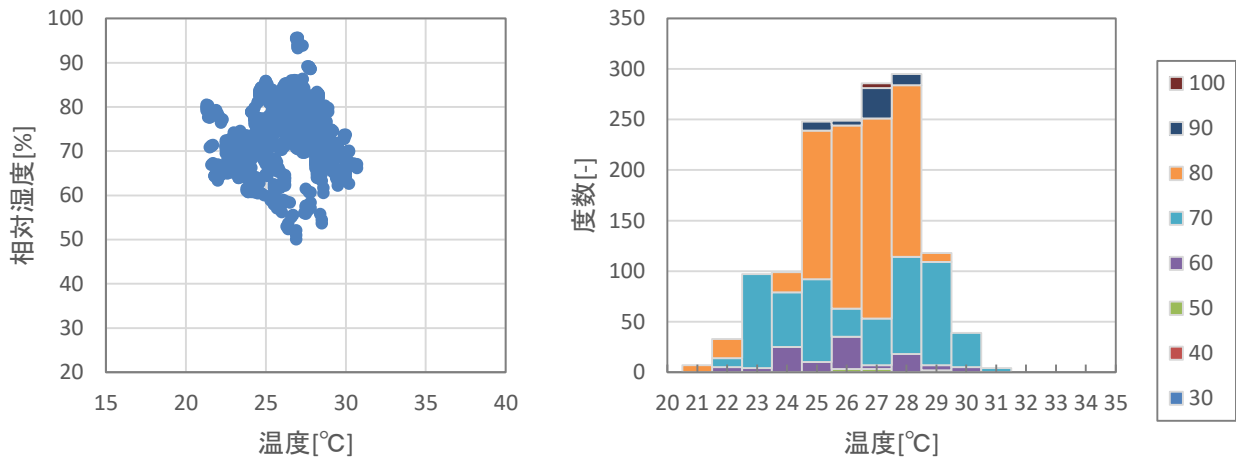


(d) 通風時の外気温度

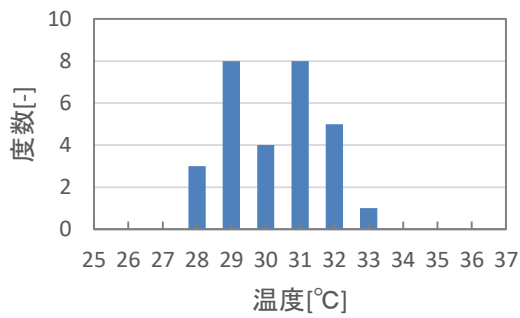
図 13 世帯 ID : 6 の寝室



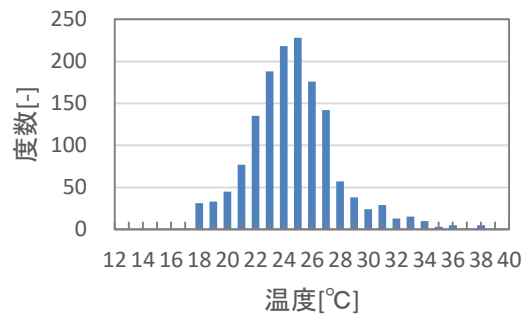
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

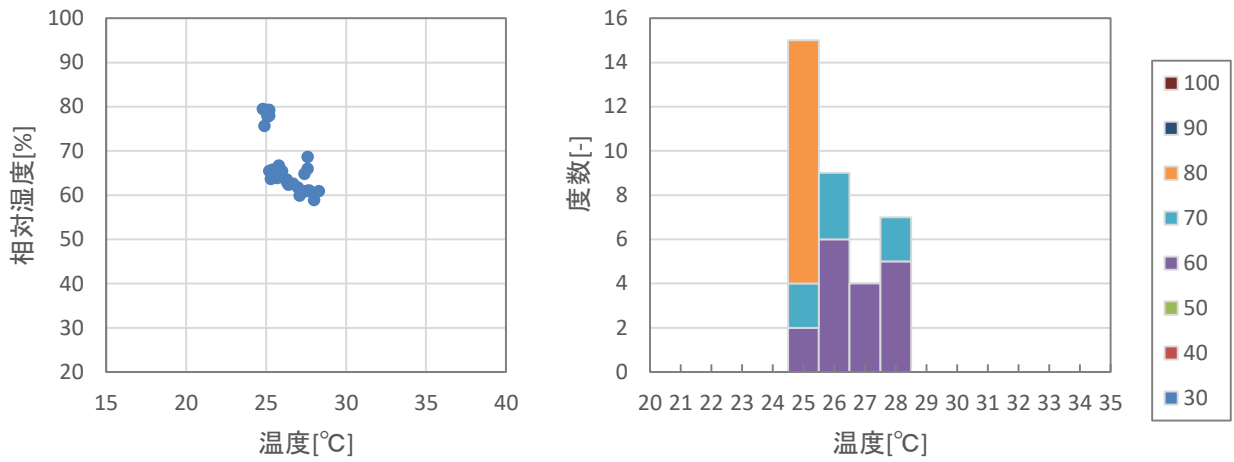


(c) 冷房開始時の室内温度

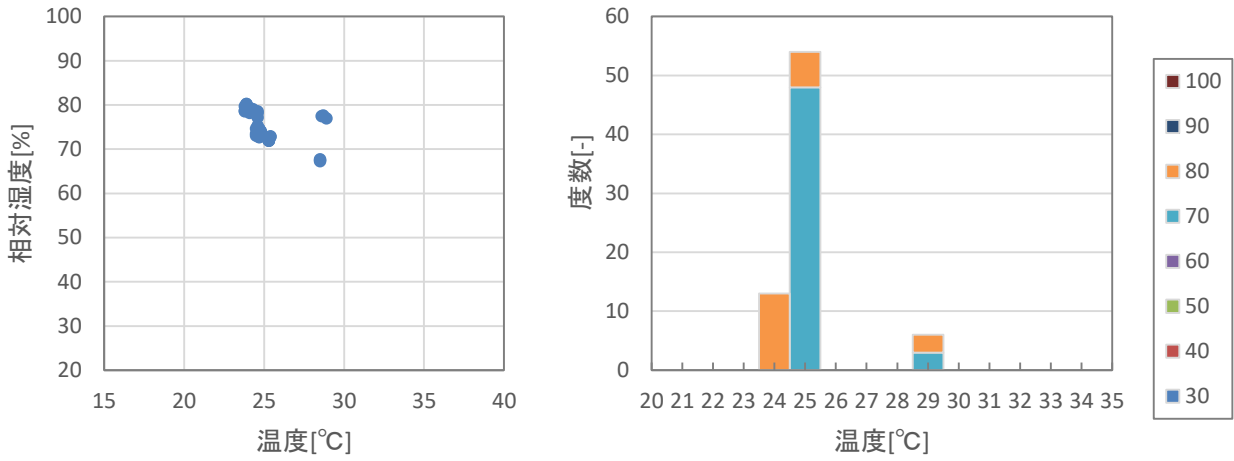


(d) 通風時の外気温度

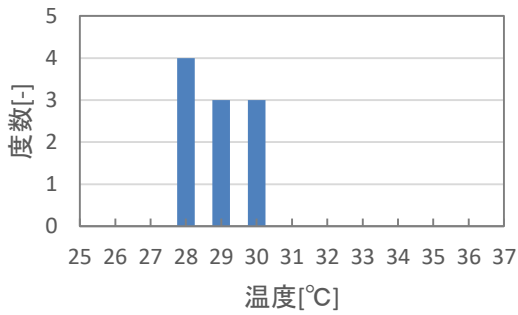
図 14 世帯 ID : 7 の居間



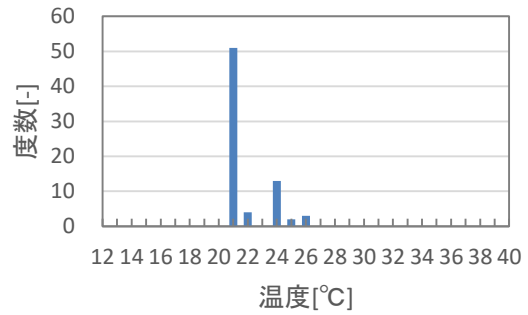
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

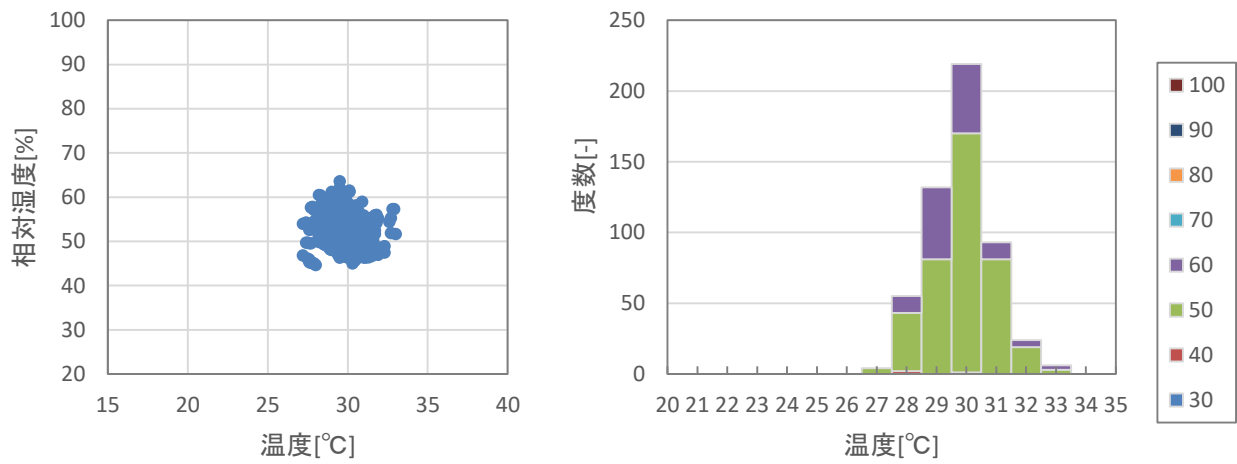


(c) 冷房開始時の室内温度

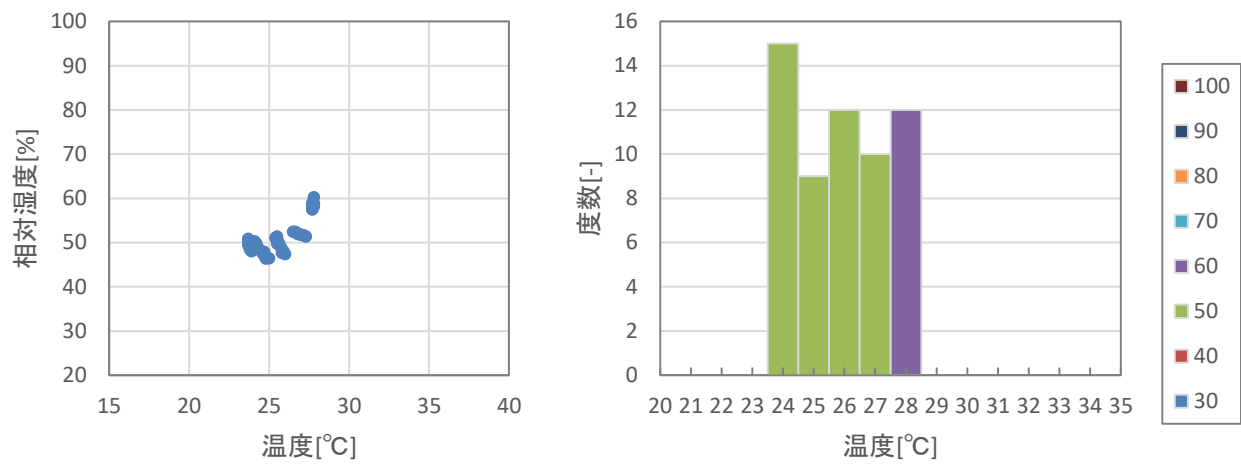


(d) 通風時の外気温度

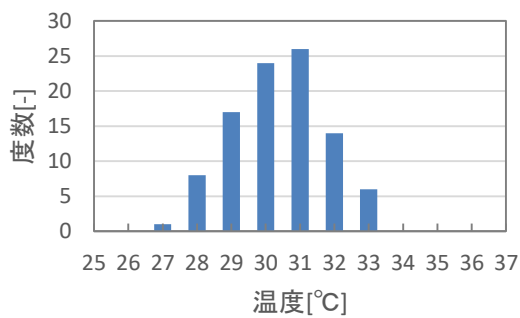
図 15 世帯 ID : 7 の寝室



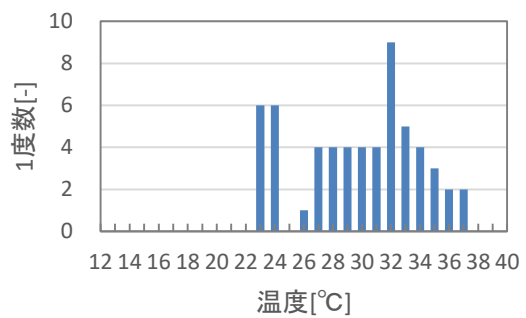
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

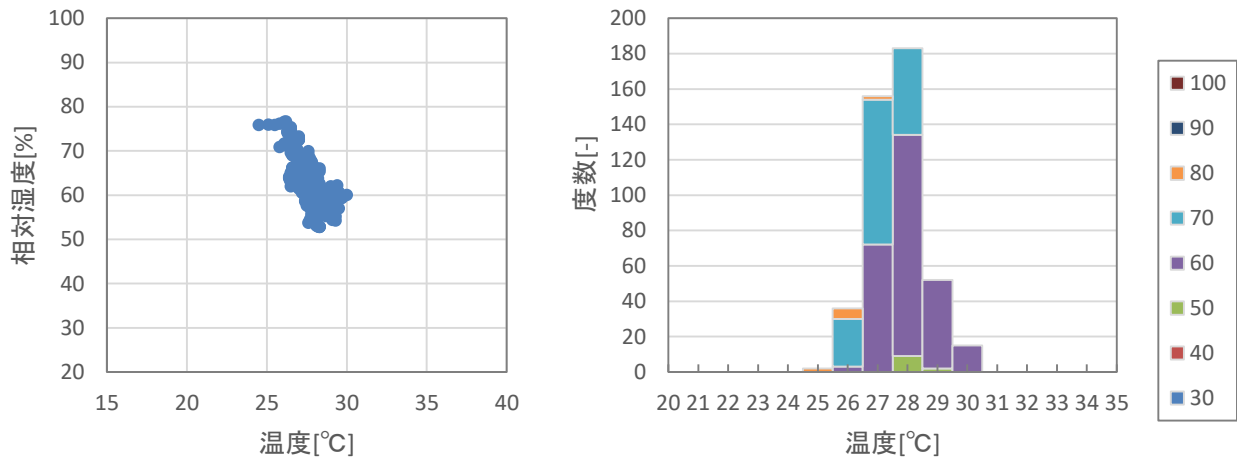


(c) 冷房開始時の室内温度

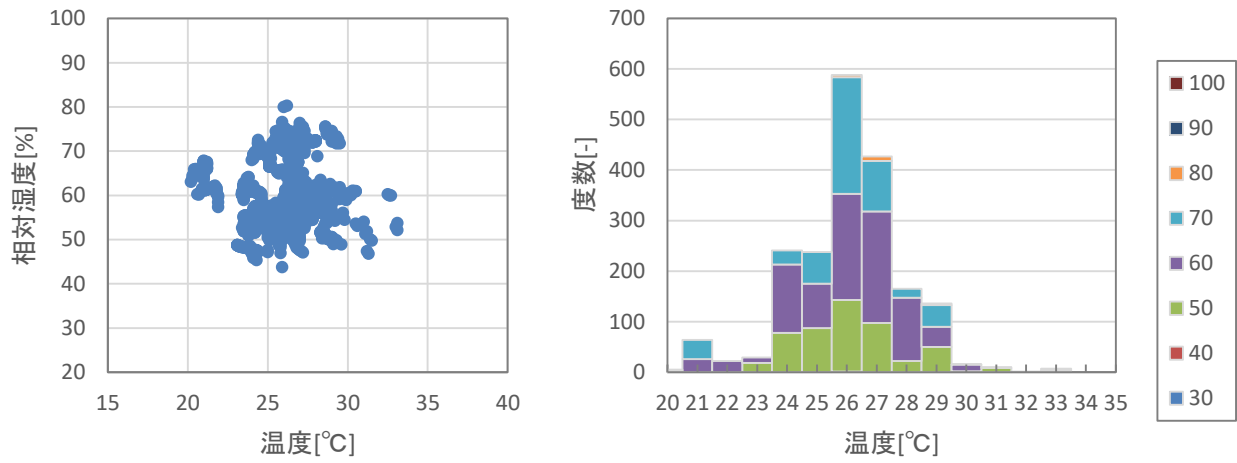


(d) 通風時の外気温度

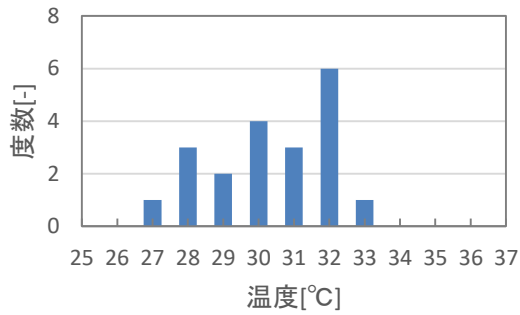
図 16 世帯 ID : 8 の居間



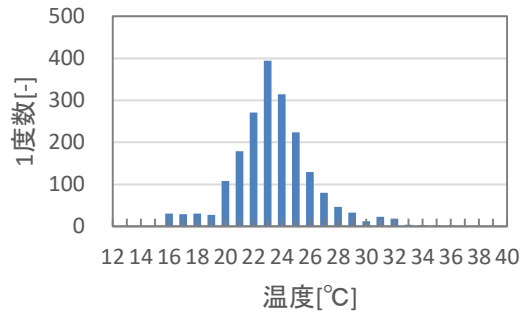
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

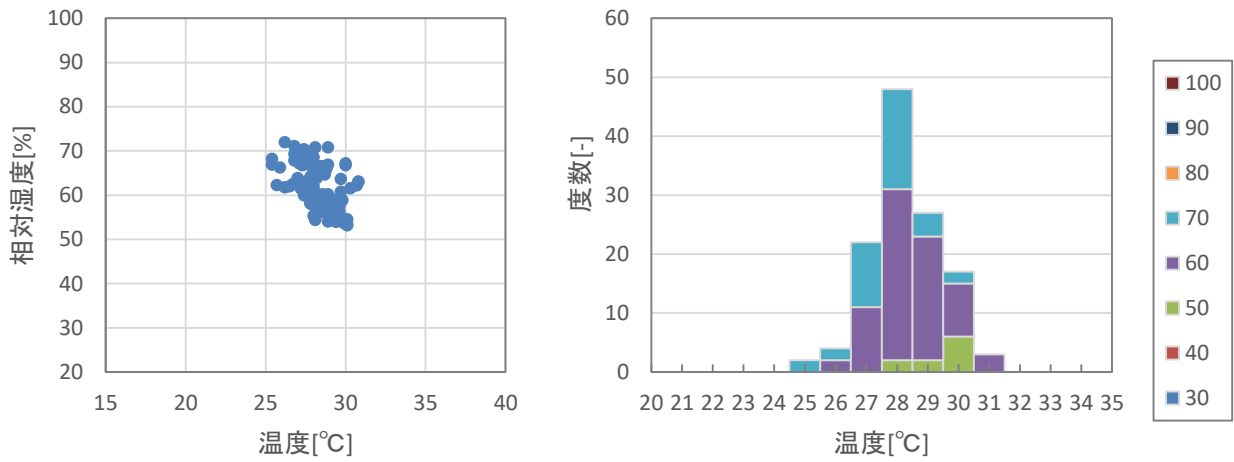


(c) 冷房開始時の室内温度

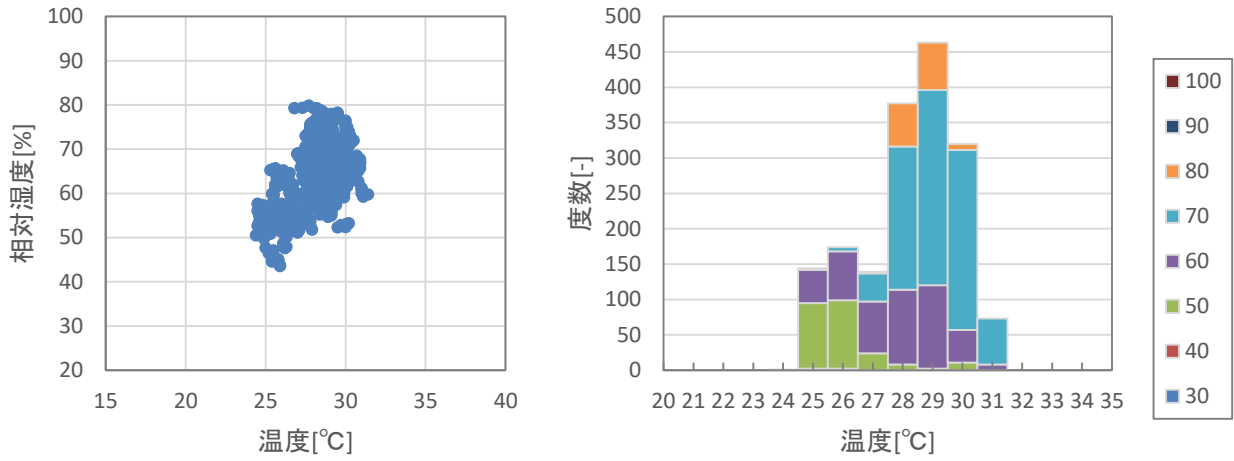


(d) 通風時の外気温度

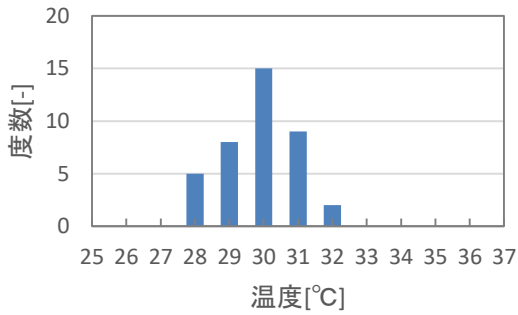
図 17 世帯 ID : 8 の寝室



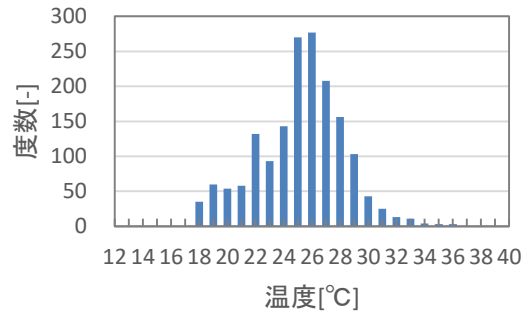
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度



(c) 冷房開始時の室内温度

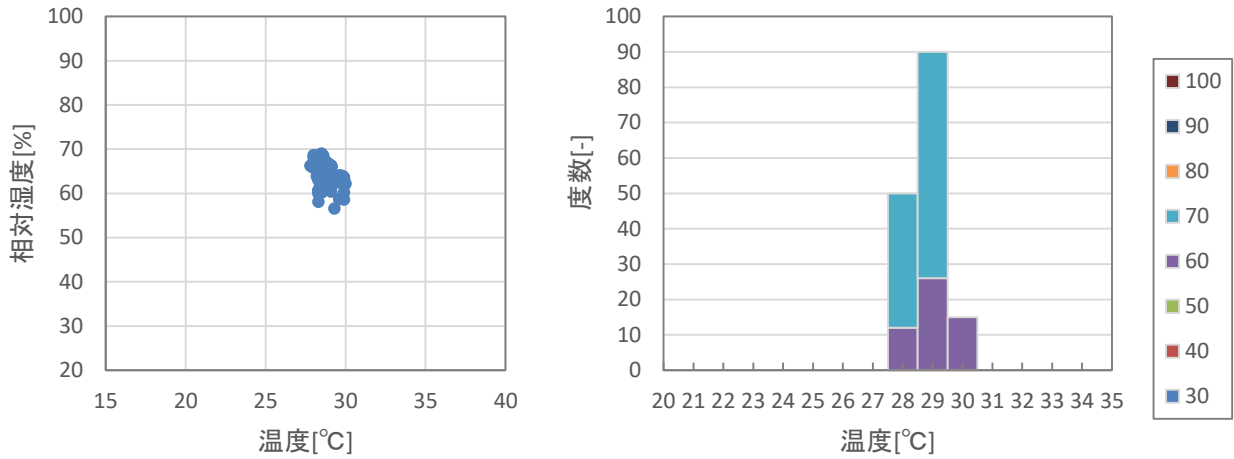


(d) 通風時の外気温度

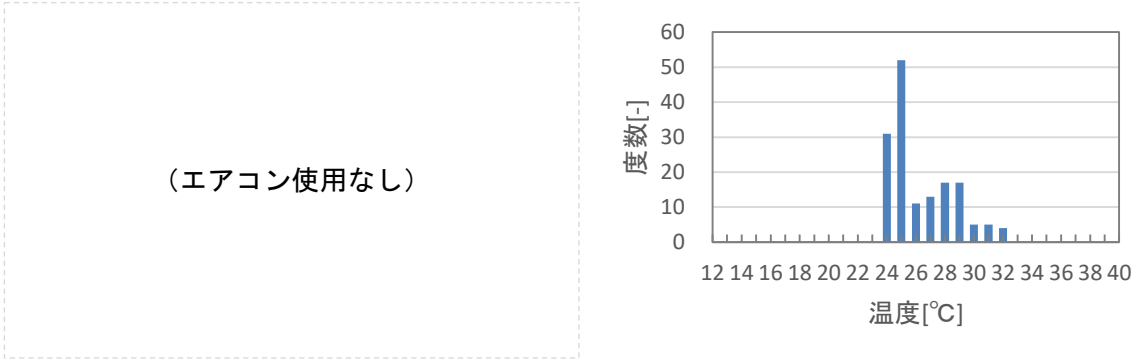
図 18 世帯 ID : 9 の居間

(エアコン使用なし)

(a) 冷房時の室内温度と相対湿度



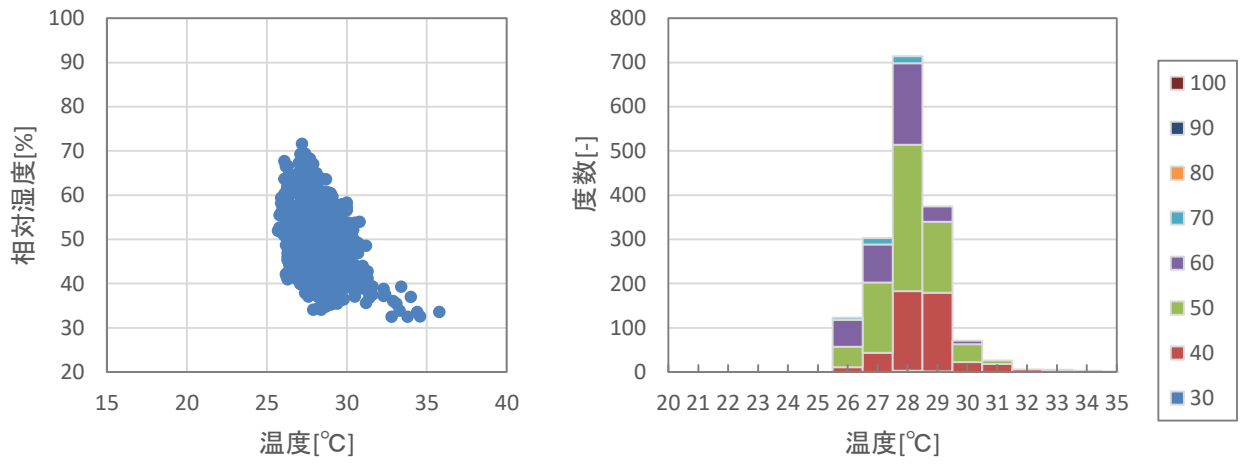
(b) 通風時の室内温度と相対湿度



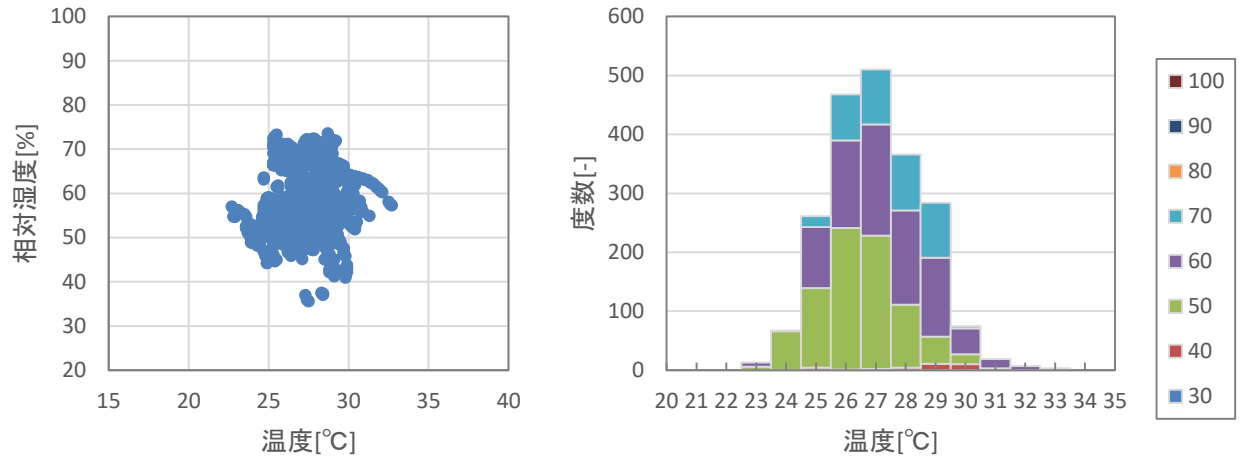
(c) 冷房開始時の室内温度

(d) 通風時の外気温度

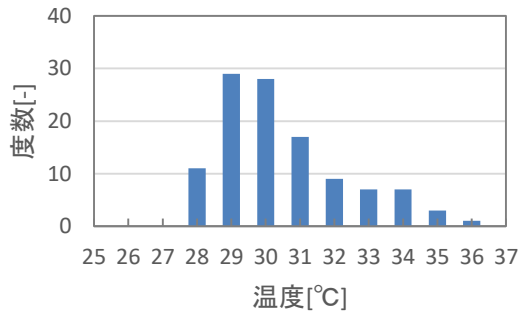
図 19 世帯 ID : 9 の寝室



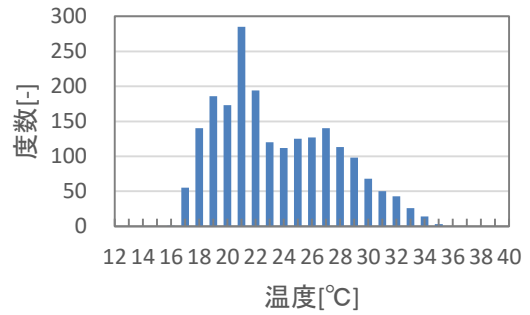
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

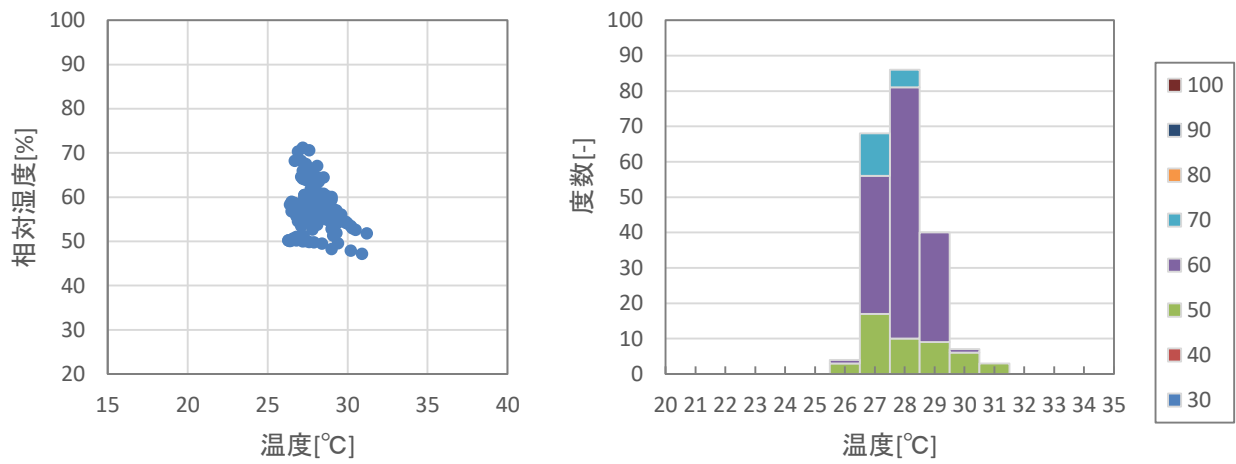


(c) 冷房開始時の室内温度

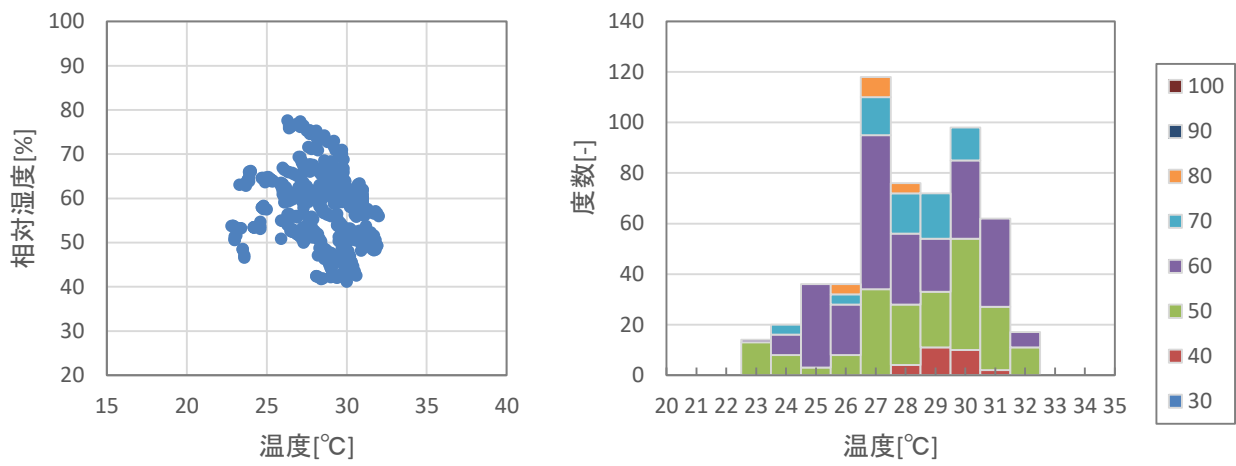


(d) 通風時の外気温度

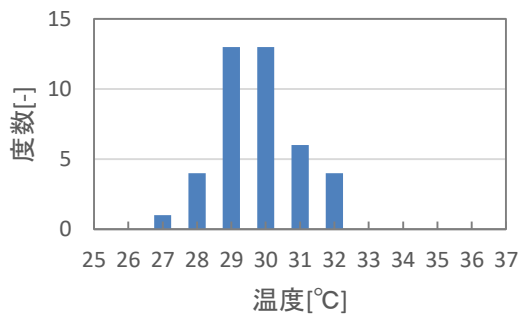
図 20 世帯 ID : 10 の居間



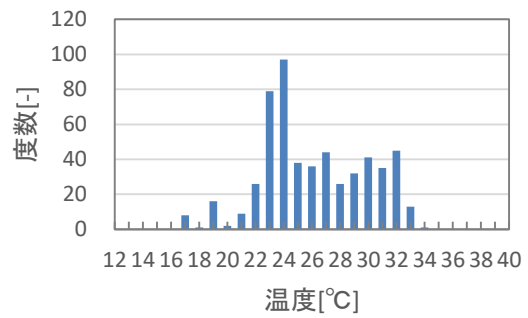
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

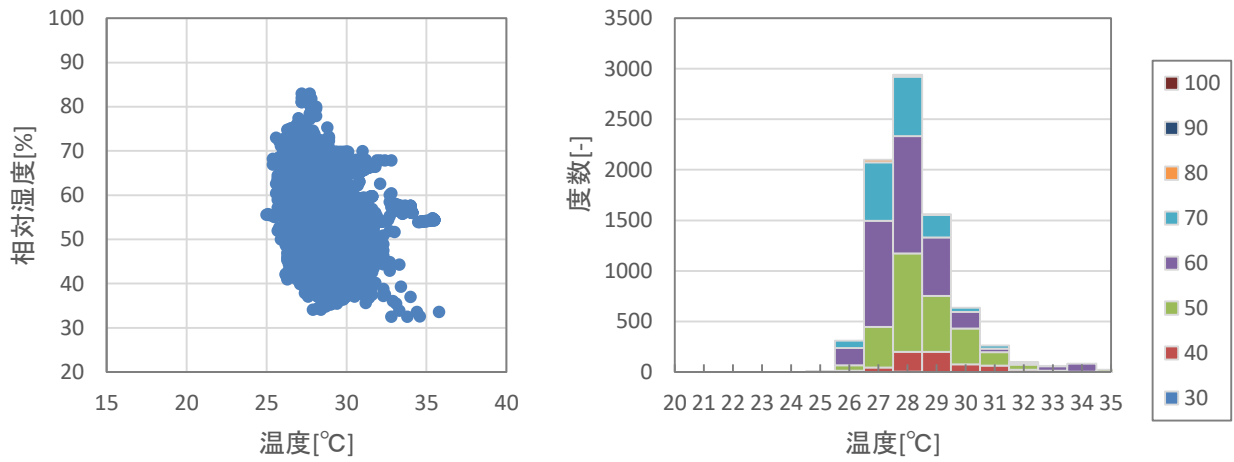


(c) 冷房開始時の室内温度

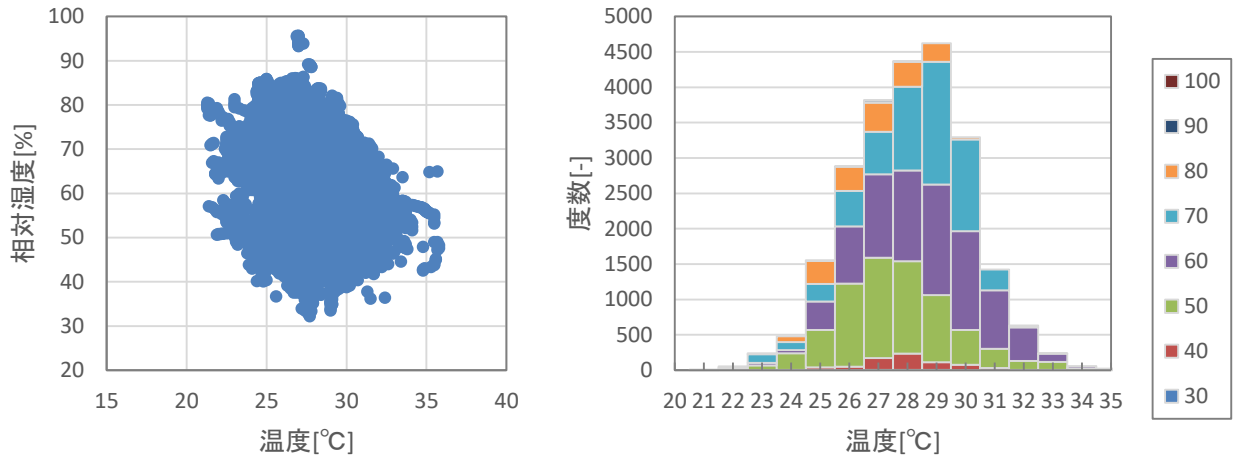


(d) 通風時の外気温度

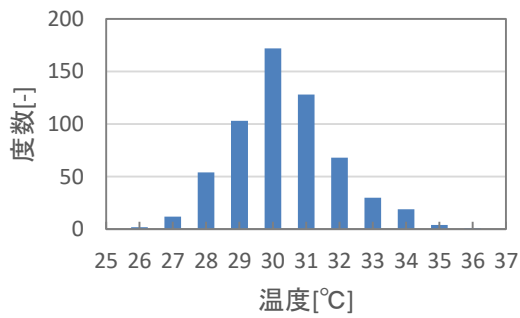
図 21 世帯 ID : 10 の寝室



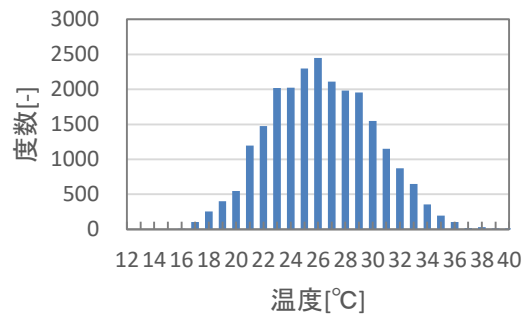
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

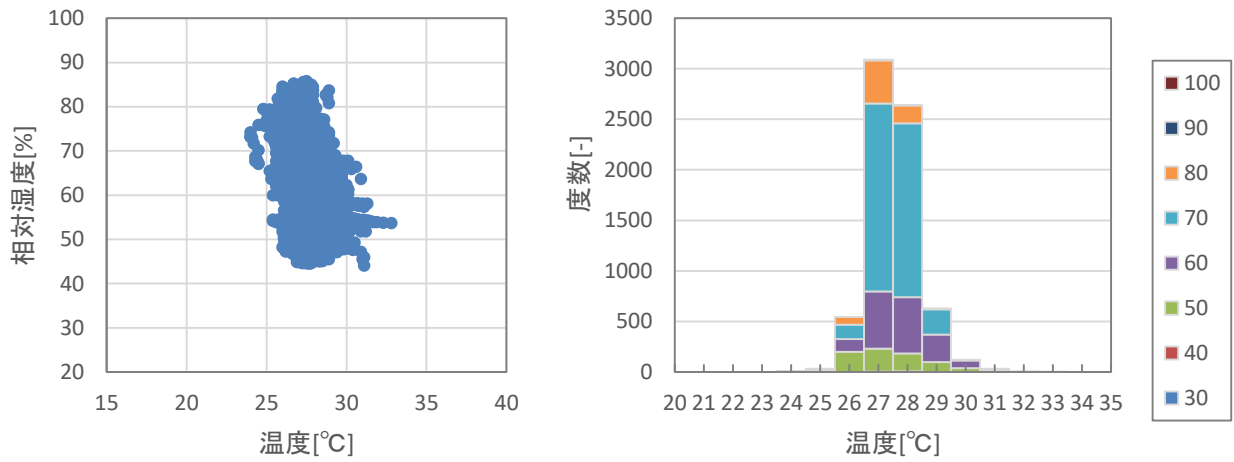


(c) 冷房開始時の室内温度

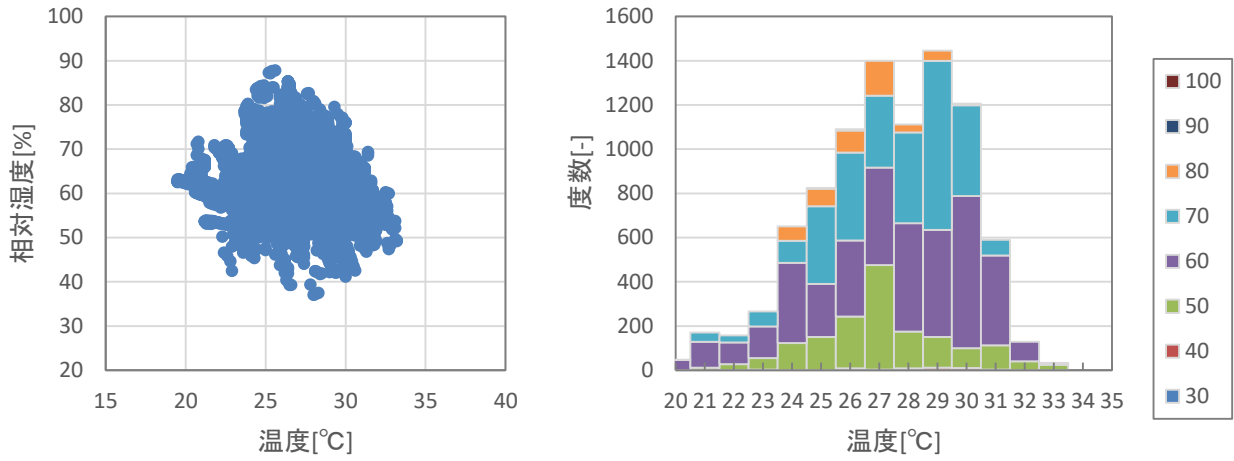


(d) 通風時の外気温度

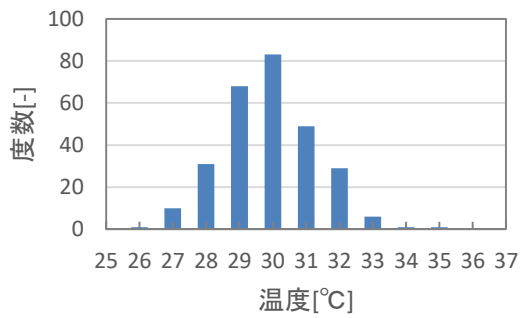
図 22 全世帯の居間



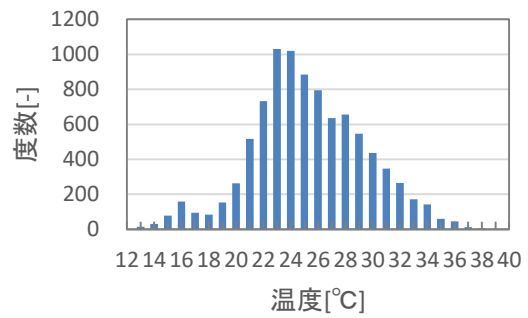
(a) 冷房時の室内温度と相対湿度



(b) 通風時の室内温度と相対湿度

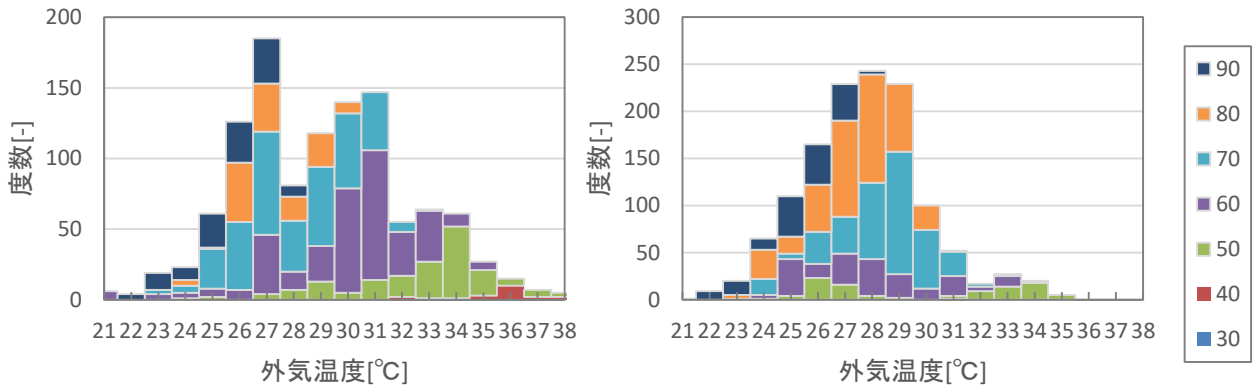


(c) 冷房開始時の室内温度

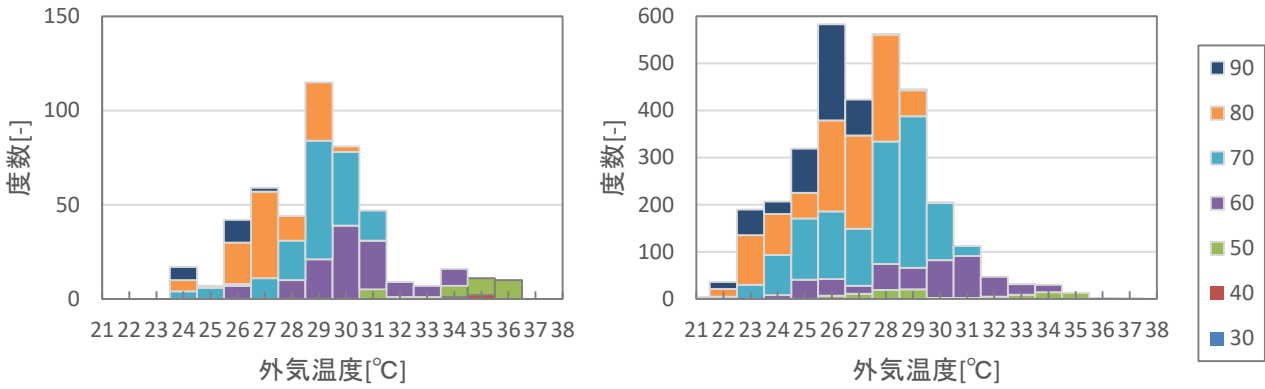


(d) 通風時の外気温度

図 23 全世帯の寝室



(a) 室内の相対湿度が60%以下の場合 (b) 室内の相対湿度が60%を超える場合
 図 24 冷房時の室内温度が $27^{\circ}\text{C} \pm 0.5^{\circ}\text{C}$ であった時の外気の温度および相対湿度 (全世帯 居間)



(a) 室内の相対湿度が60%以下の場合 (b) 室内の相対湿度が60%を超える場合
 図 25 冷房時の室内温度が $27^{\circ}\text{C} \pm 0.5^{\circ}\text{C}$ であった時の外気の温度および相対湿度 (全世帯 寝室)

付録 D

蒸暑地実住宅におけるエアコン冷房挙動の実測調査

1 蒸暑地実住宅におけるエアコン冷房挙動の実測調査

1.1 目的

蒸暑地におけるルームエアコンの使用状況と空調対象室の温湿度を測定し、温熱環境と機器効率についての問題点、特に除湿の観点からのものを把握する事を目的とする。

1.2 実測概要

蒸暑地の単身者世帯においてルームエアコンの使用状況と空調対象室の温湿度を測定する。蒸暑地の住宅として沖縄県宮古市にある1階建てRC造を実測対象住宅とした。ルームエアコンの使用状況を把握するために機器全体の消費電力、室外機と室内機の冷媒温度、吸込みと吹き出しの空気温湿度を測定する。空調対象室の温熱環境を把握するために高さ方向の温度分布と床面から1200 mm付近の相対湿度を測定する。

1.2.1 対象住宅

1.2.1.1 戸建住宅

測定対象住宅の概要を以下に示す。居住者は単身の高齢者である。

- 所在地 : 沖縄県宮古市
- 構造 : RC造 (壁構造)
- 階数 : 1階
- 延床面積 : 129.93 m²
- 竣工年 : 平成19年4月

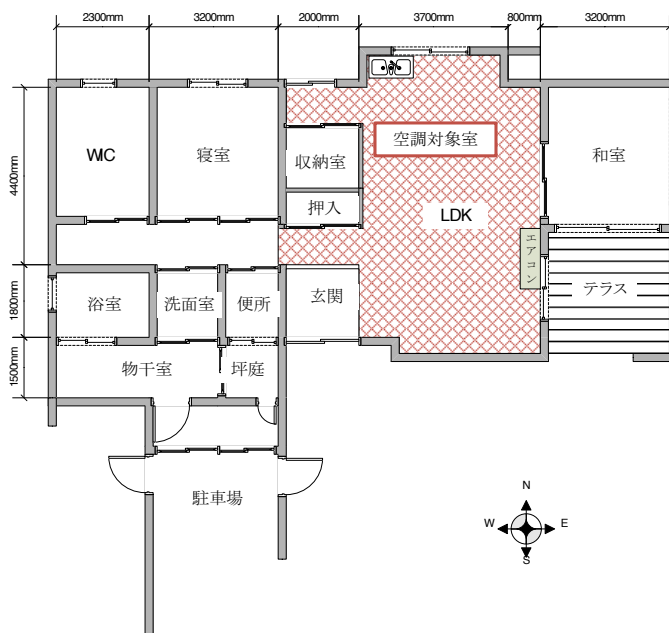


図 1 平面図



图 2 立面图 (南)

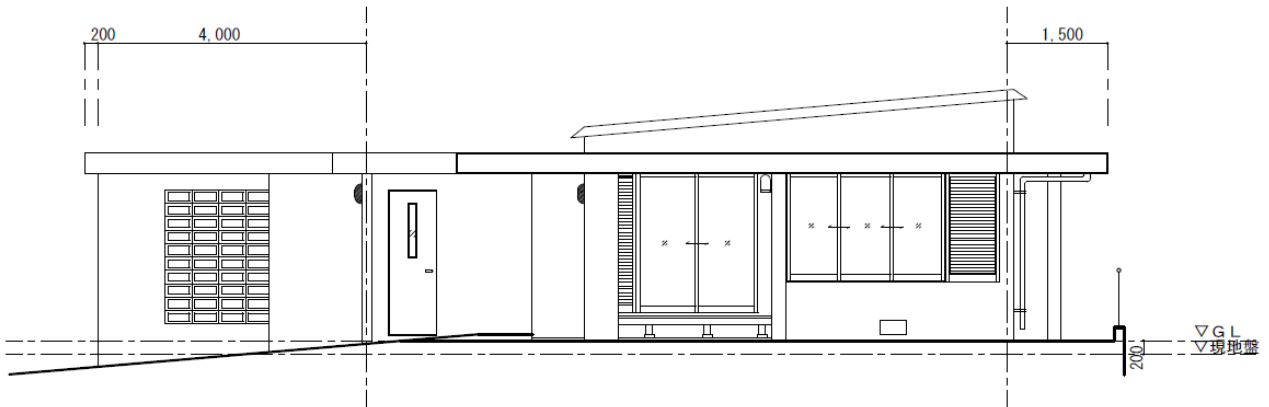


图 3 立面图 (東)

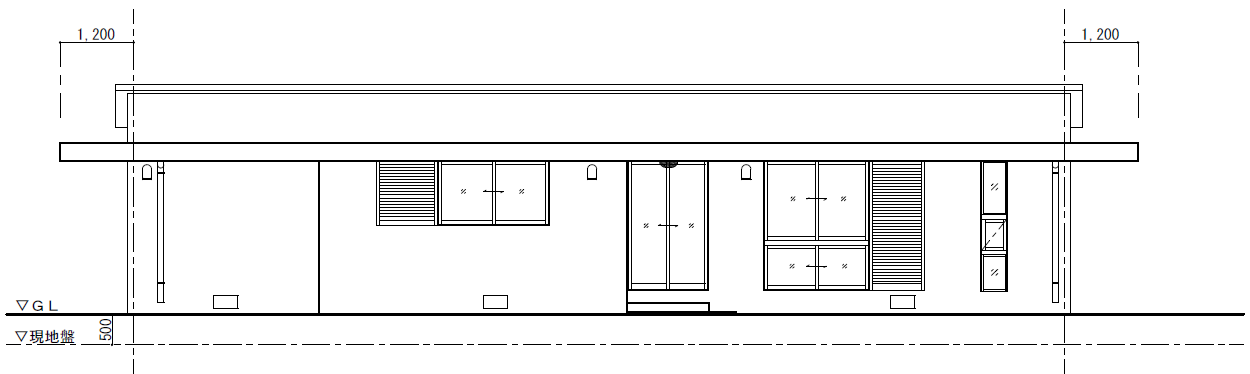


图 4 立面图 (北)

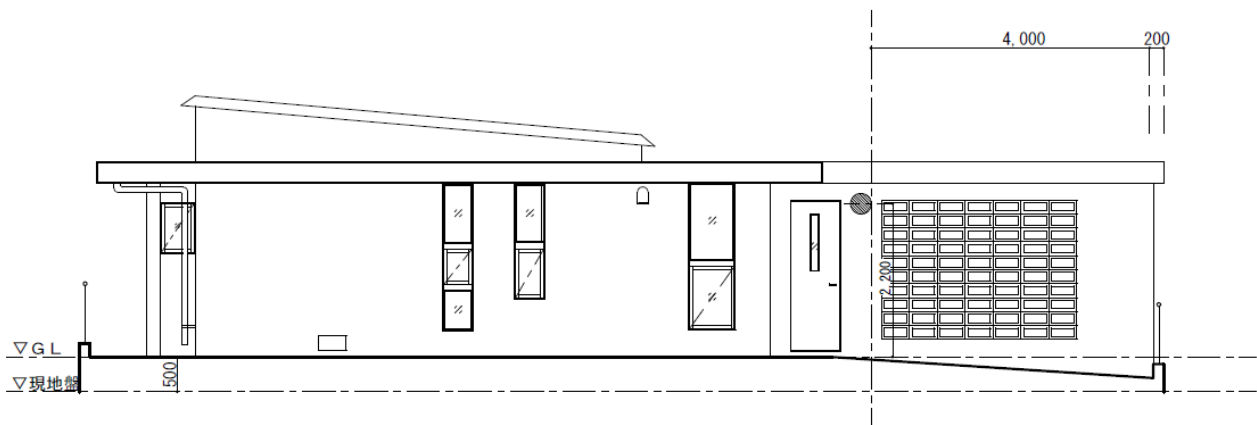


図 5 立面図 (西)

1.2.1.2 集合住宅

測定対象住宅の概要を以下に示す。居住者は1家族（大人2人、子供2人、幼児1人）である。

- 所在地 : 沖縄県宮古市
- 構造 : RC造（壁構造）
- 階数 : 6階（対象住宅は3階の住戸）
- 延床面積 : 51.03 m²

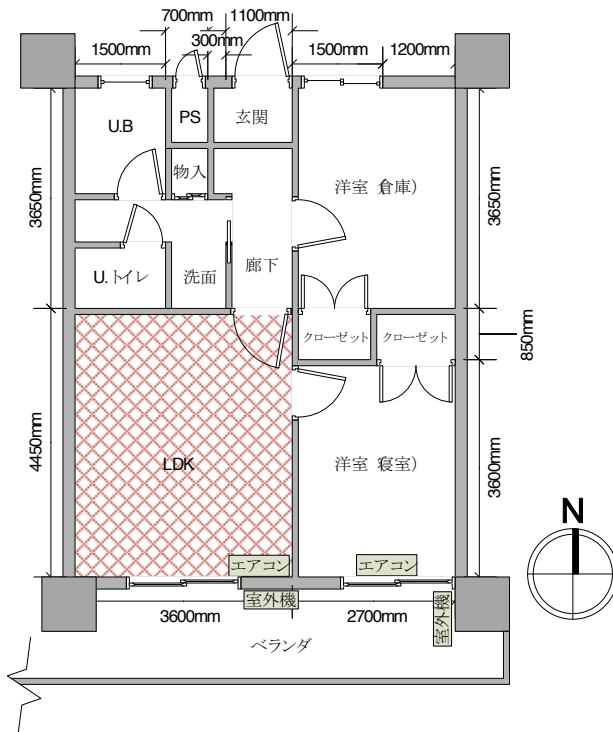


図 6 平面図

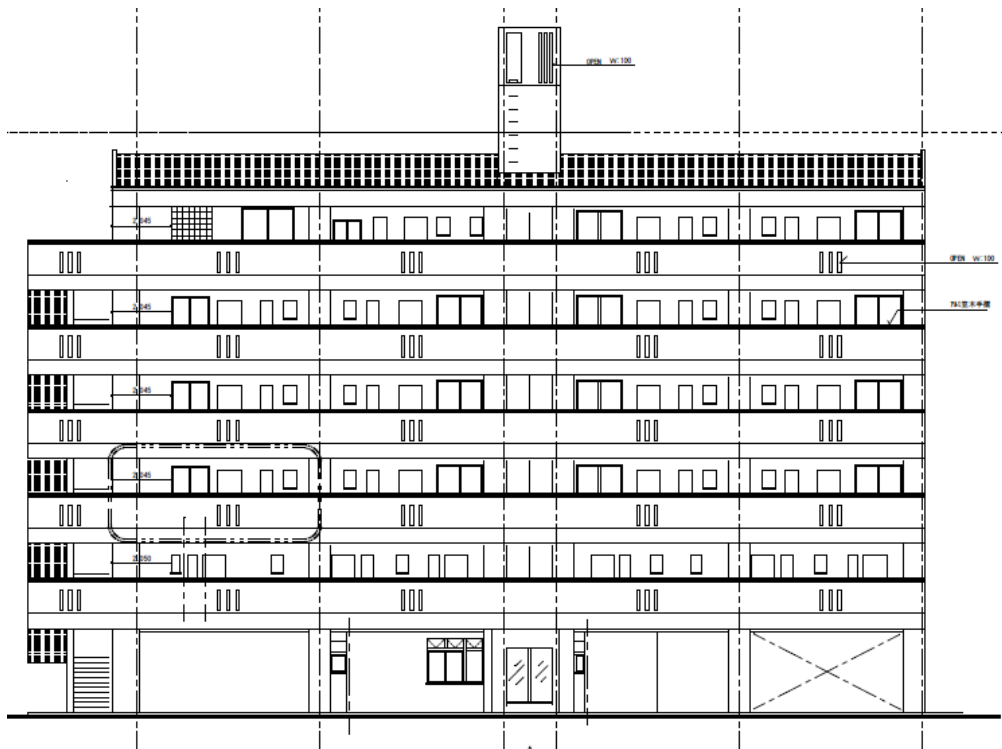


图 7 立面图 (南)

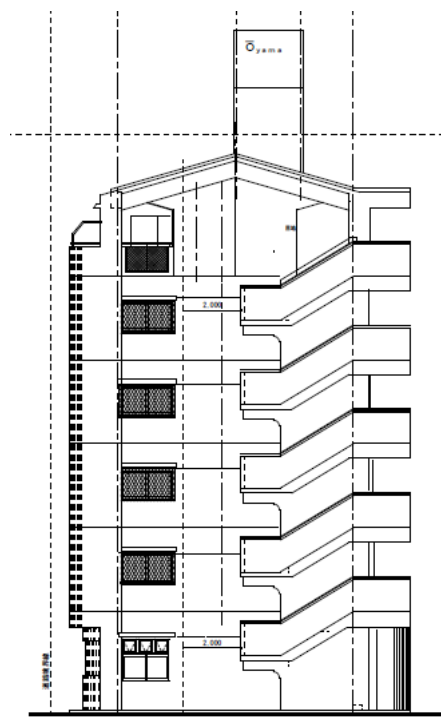


图 8 立面图 (东)

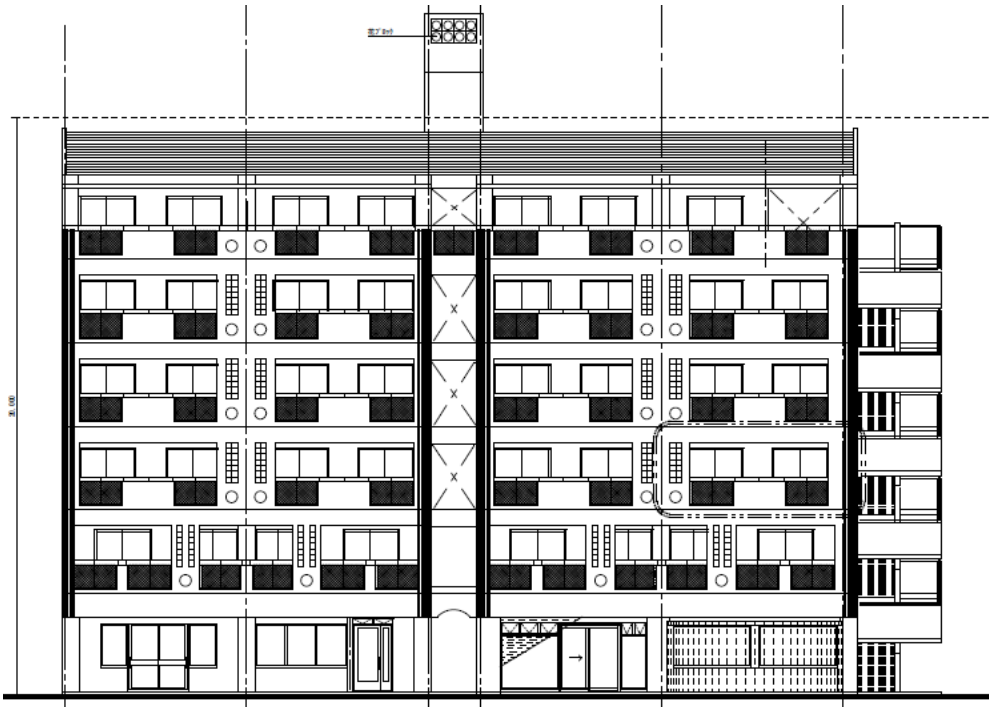


图 9 立面图 (北)

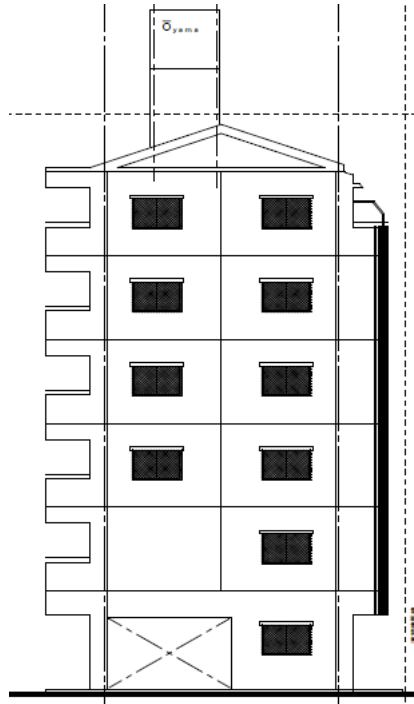


图 10 立面图 (西)

1.2.2 エアコン仕様

測定に使用するエアコンの仕様を以下に示す。

冷房定格能力 : 6.3 kW
 定格消費電力 : 1.92 kW

1.2.3 測定項目

測定項目	対象	測定機器	補足
電力	ルームエアコン全体	CT	
冷媒温度	室外機 圧縮機 入口	T型熱電対	
	室外機 圧縮機 出口		
	室外機 熱交換器 入口		
	室外機 熱交換器 中央		
	室外機 熱交換器 出口		
	室内機 膨張弁 入口		
	室内機 膨張弁 出口		
	室内機 熱交換器 入口		
	室内機 熱交換器 中央1		
	室内機 熱交換器 中央2		
	室内機 熱交換器 出口		
空気温度	室外機 吸い込み	T型熱電対	
	室外機 吹き出し		
	室内機 吸い込み	白金抵抗式	
	室内機 吹き出し 左		
	室内機 吹き出し 中央		
	室内機 吹き出し 右		
空気湿度	室外機 吸い込み	抵抗変化式	
	室外機 吹き出し		
	室内機 吸い込み	静電容量式	
	室内機 吹き出し 中央		
回転数	室内機 送風ファン	レーザー回転数計	
	室外機 圧縮機 回転数	電流クランプ	

測定項目	対象	測定機器	補足
空気温度	床面からの高さ 50 mm	温湿度ロガー	居住者の負担にならないよう既設の家具類を利用して設置する
	床面からの高さ 600 mm		
	床面からの高さ 900 mm		
	床面からの高さ 1,200 mm		
	床面からの高さ 1,800 mm		
	床面からの高さ 2,400 mm		
空気湿度	床面からの高さ 1,200 mm		

1.2.4 測定期間

1.2.4.1 戸建て住宅

2014年4月～2014年11月

1.2.4.2 集合住宅

2015年4月～2015年11月

1.3 実測結果

1.3.1 戸建住宅

1.3.1.1 月別使用実態

月別使用実態を時系列グラフで以下に示す。戸建て住宅では測定期間を通して日中は通風を行うためエアコンを使用せず夜間に使用する傾向が見られた。

(1) 4月

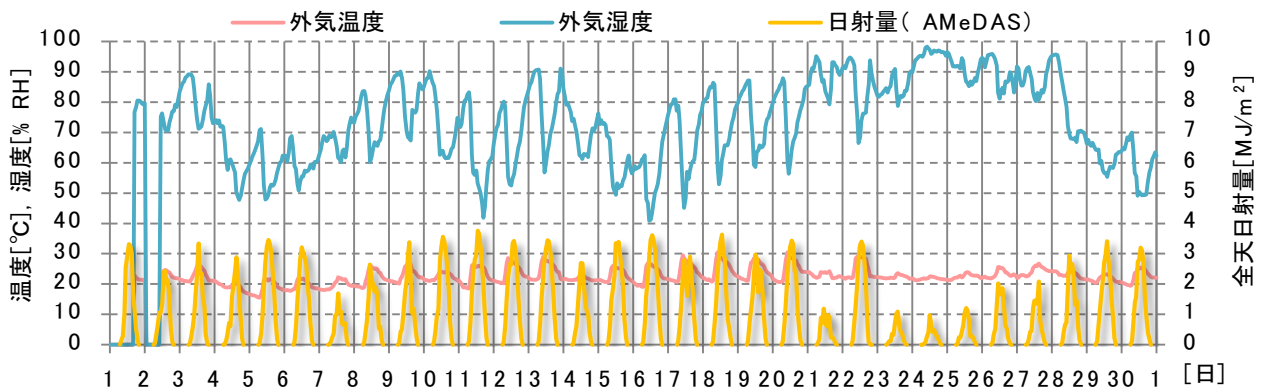


図 11 外気条件 (2014年4月)

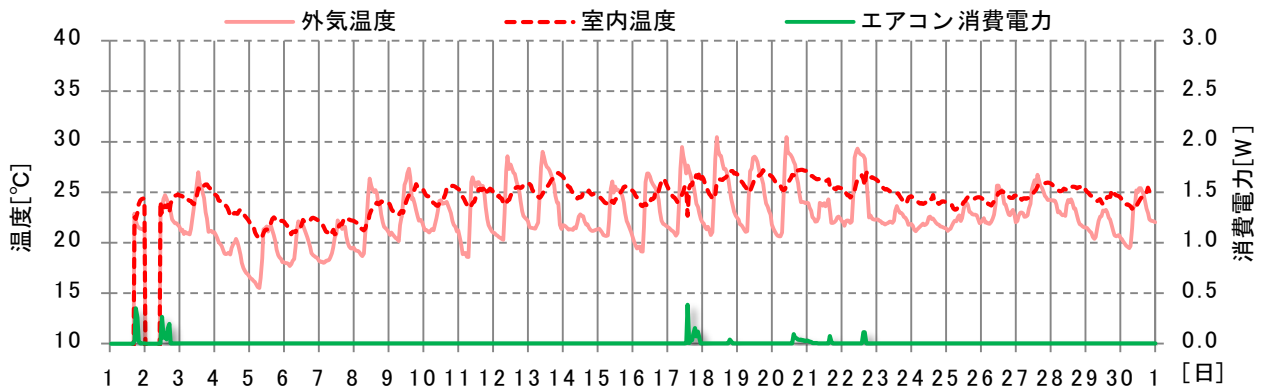


図 12 エアコン稼働状態と室内外温度 (2014年4月)

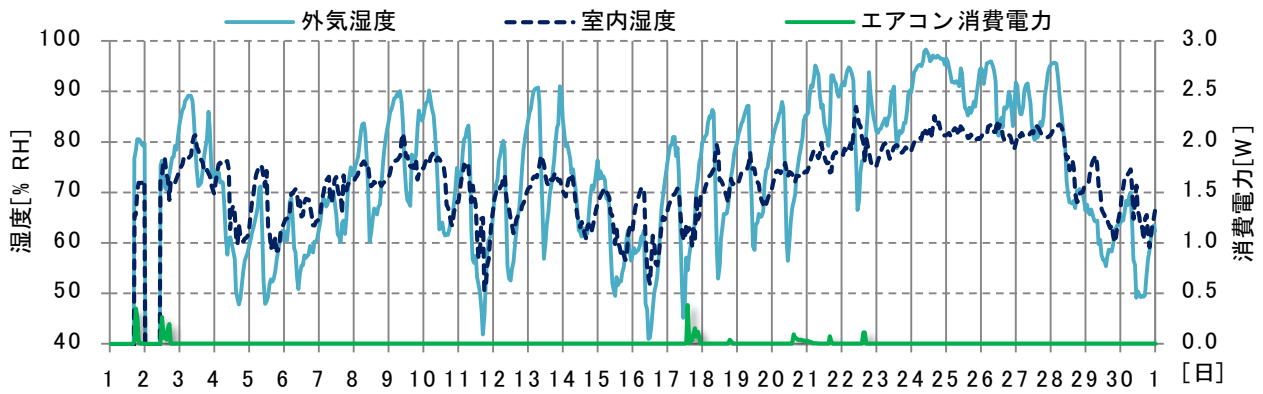


図 13 エアコン稼働状態と室内外湿度 (2014 年 4 月)

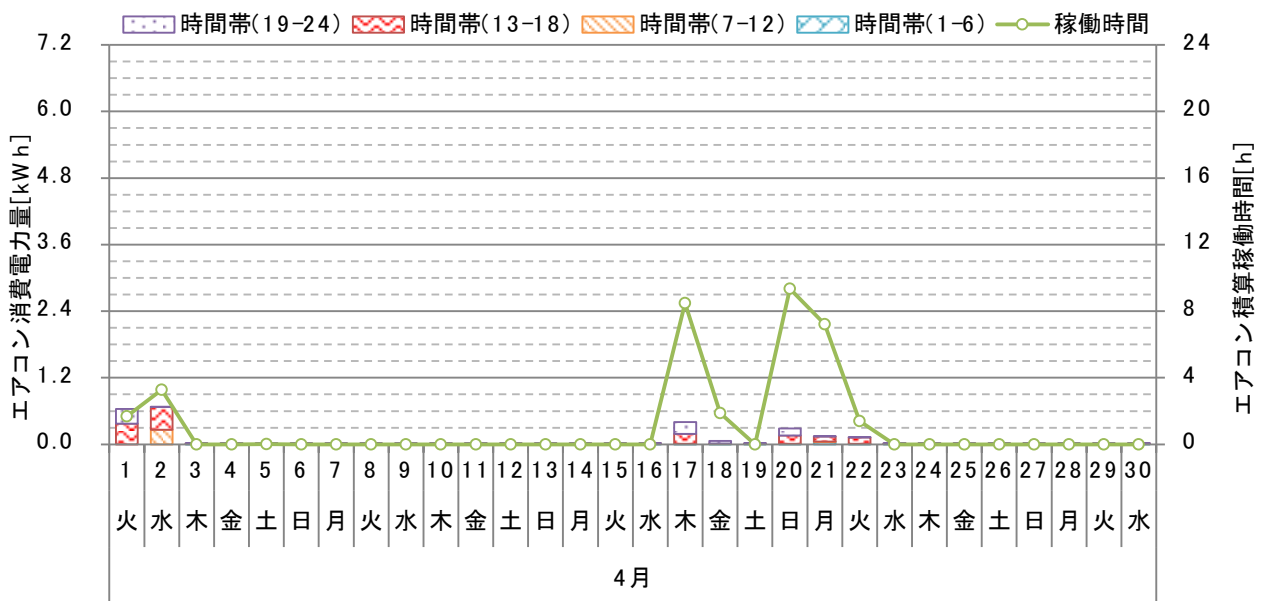


図 14 日別の時間帯別消費電力量と積算稼働時間 (2014 年 4 月)

(2) 5 月

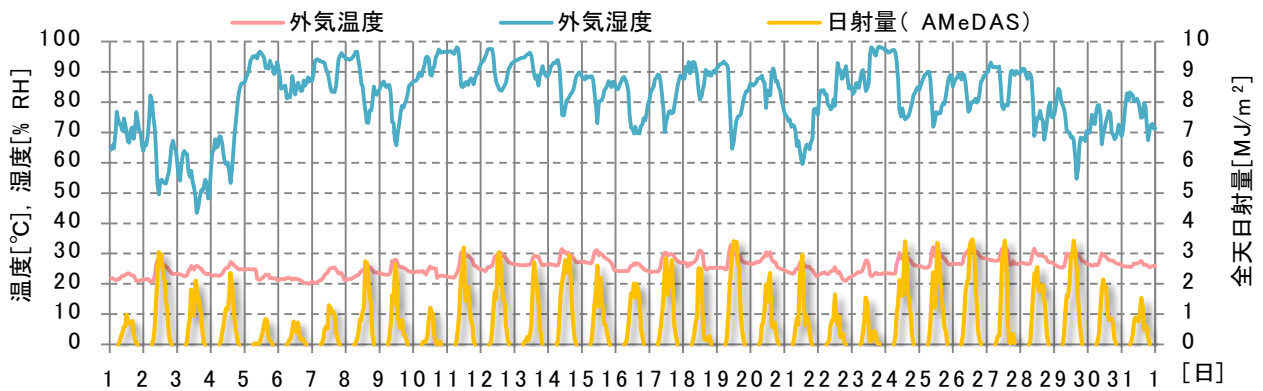


図 15 外気条件 (2014 年 5 月)

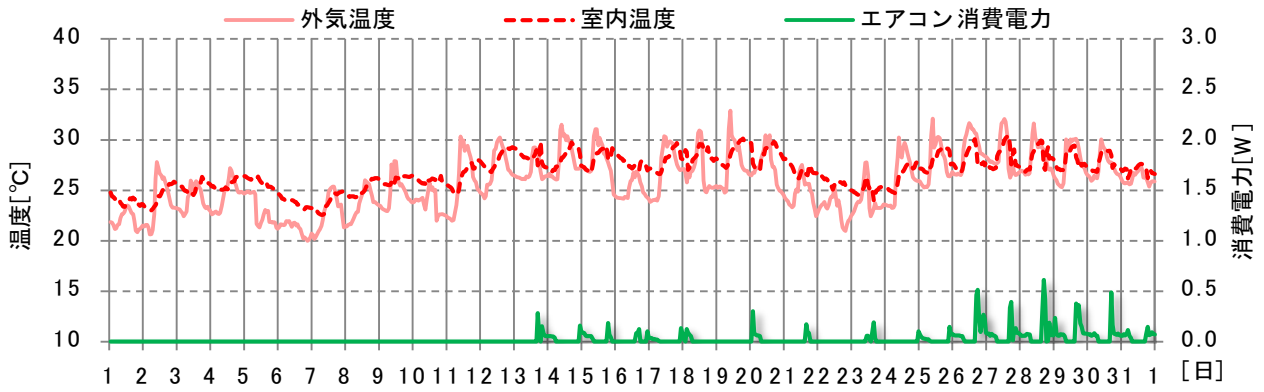


図 16 エアコン稼働状態と室内外温度 (2014 年 5 月)

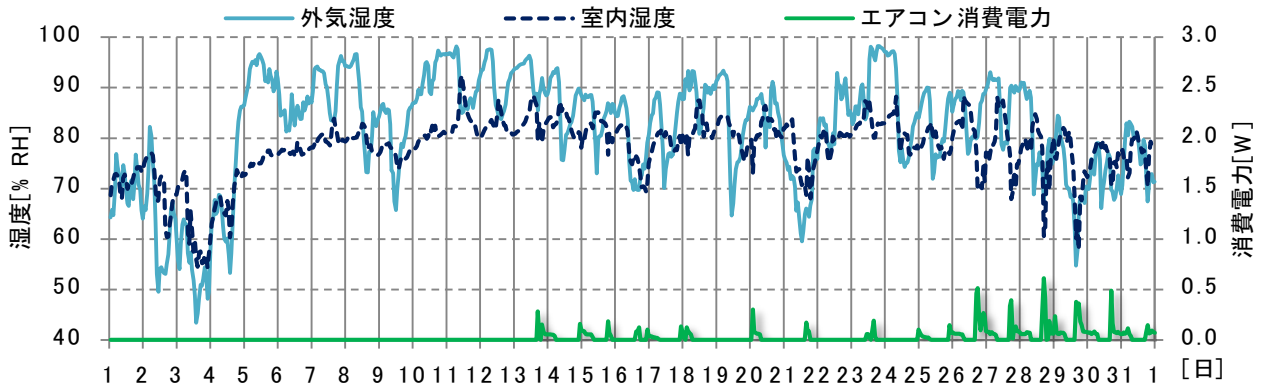


図 17 エアコン稼働状態と室内外湿度 (2014 年 5 月)

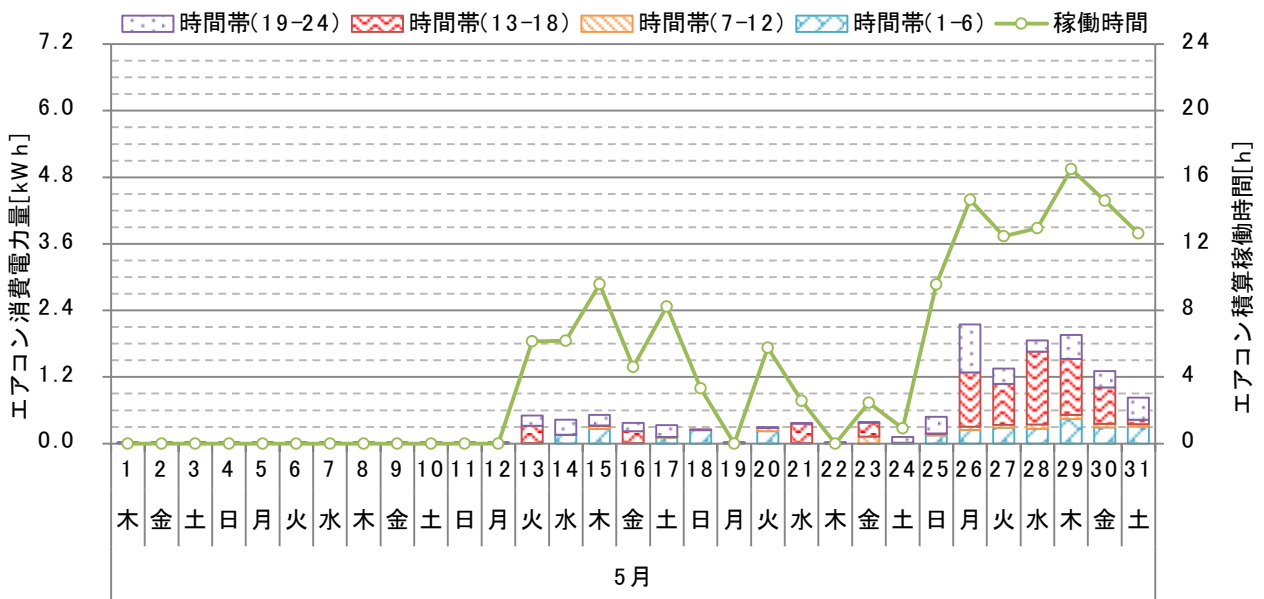


図 18 日別の時間帯別消費電力量と積算稼働時間 (2014 年 5 月)

(3) 6月

外気温センサーが鼠害によって故障したため AMeDAS で代用した。

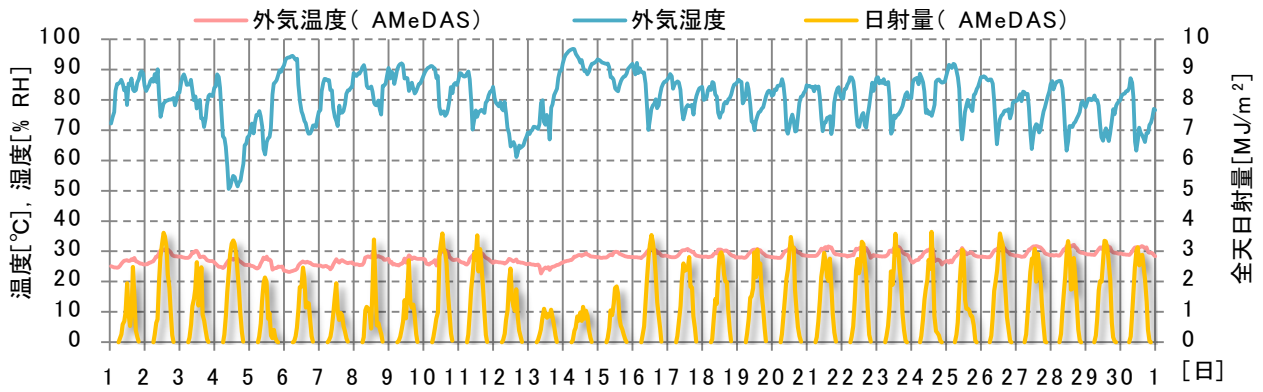


図 19 外気条件 (2014 年 6 月)

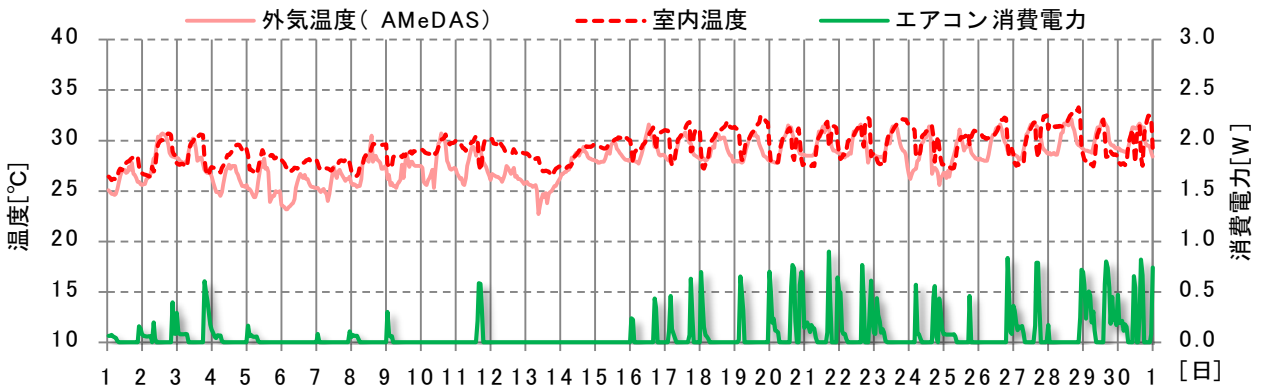


図 20 エアコン稼働状態と室内外温度 (2014 年 6 月)

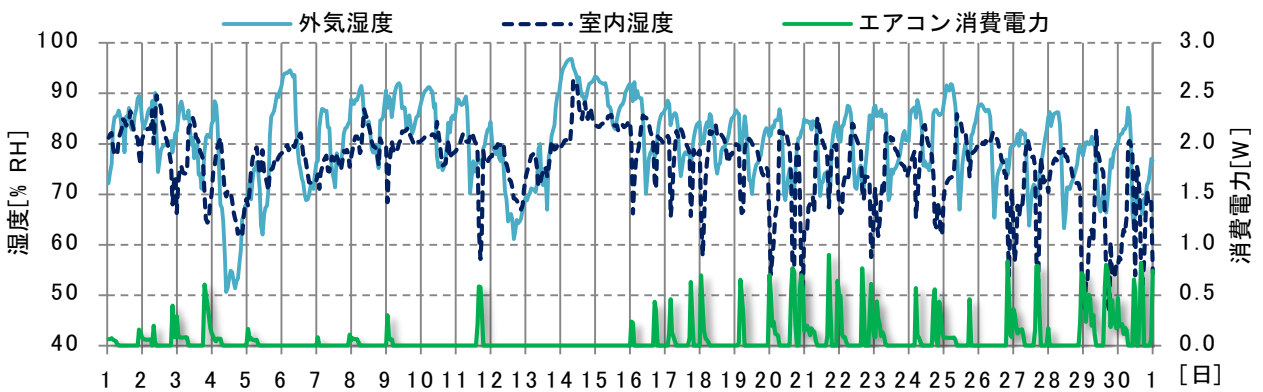


図 21 エアコン稼働状態と室内外湿度 (2014 年 6 月)

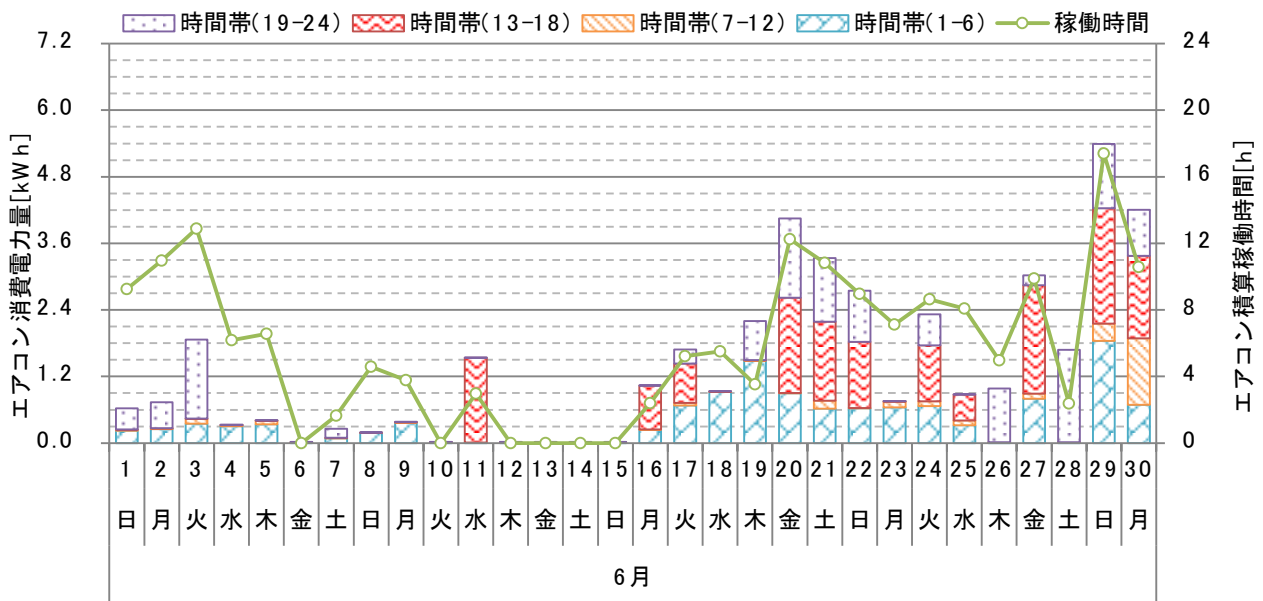


図 22 日別の時間帯別消費電力量と積算稼働時間 (2014 年 6 月)

(4) 7 月

外気温度センサーが鼠害によって故障したため AMeDAS で代用した。

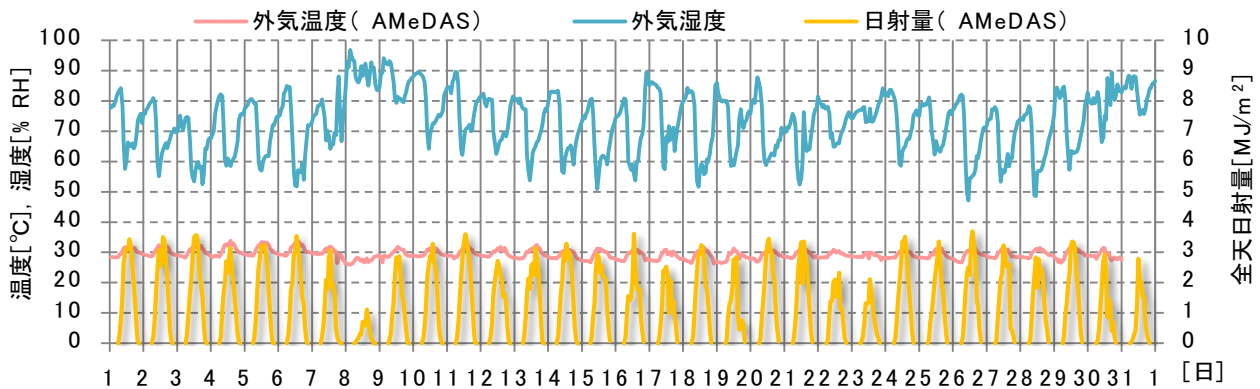


図 23 外気条件 (2014 年 7 月)

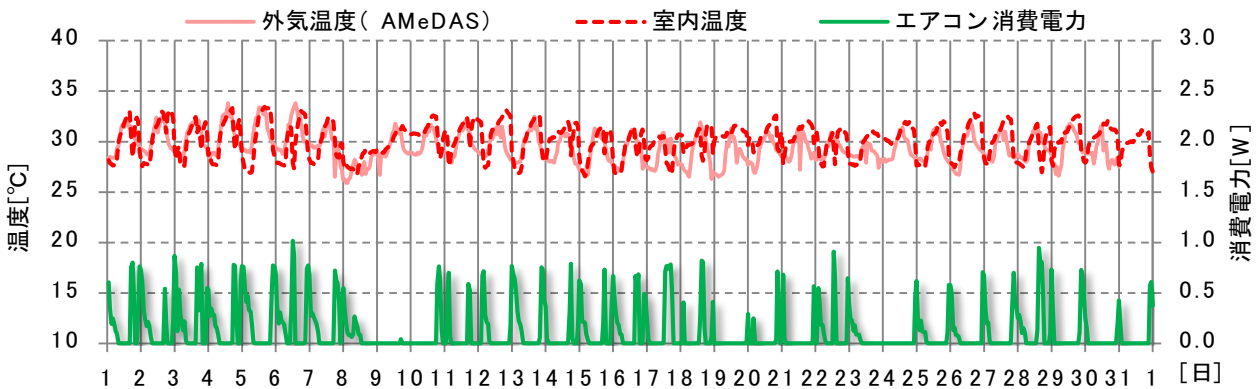


図 24 エアコン稼働状態と室内外温度 (2014 年 7 月)

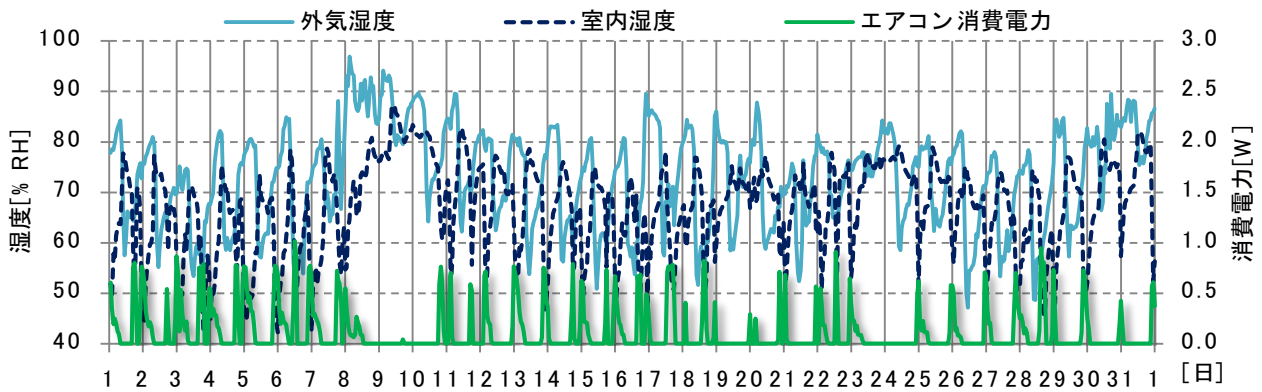


図 25 エアコン稼働状態と室内外湿度 (2014年7月)

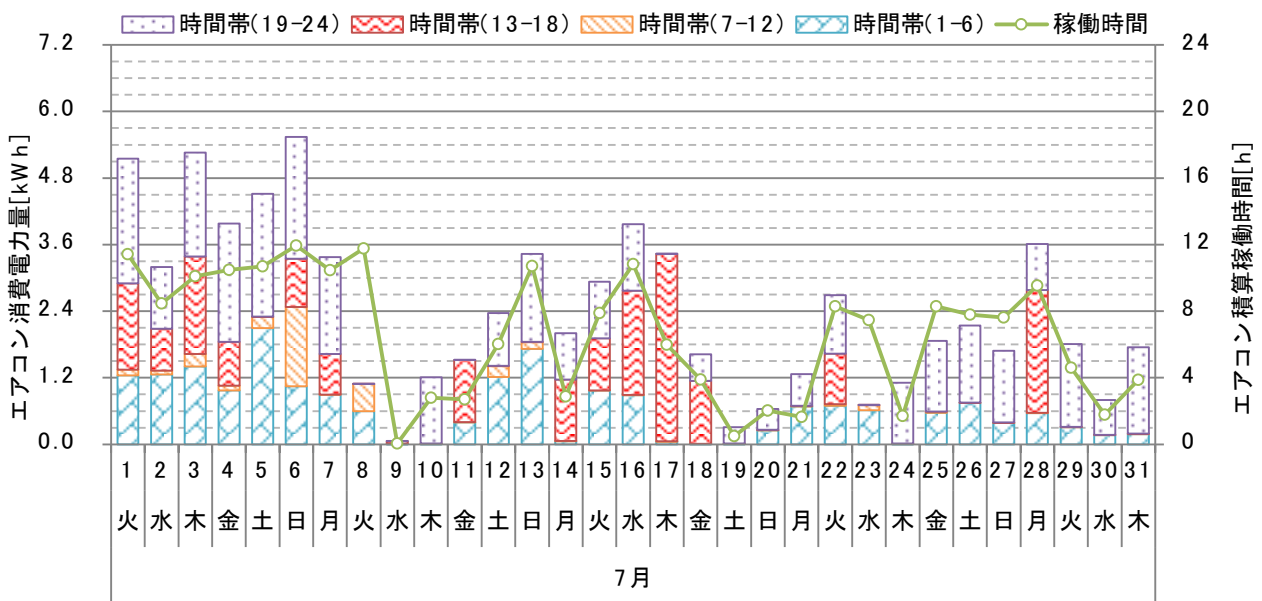


図 26 日別の時間帯別消費電力量と積算稼働時間 (2014年7月)

(5) 8月

外気温度センサーが鼠害によって故障したため AMeDAS で代用した。

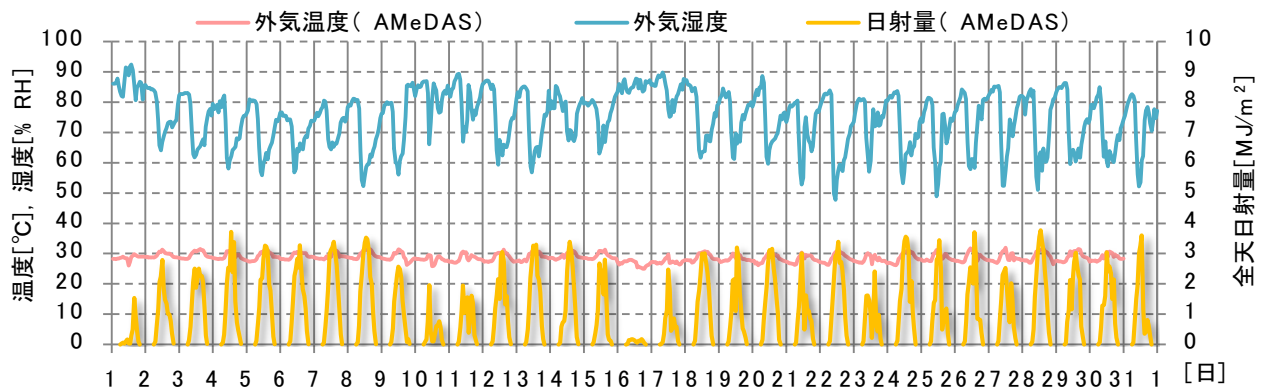


図 27 外気条件 (2014年8月)

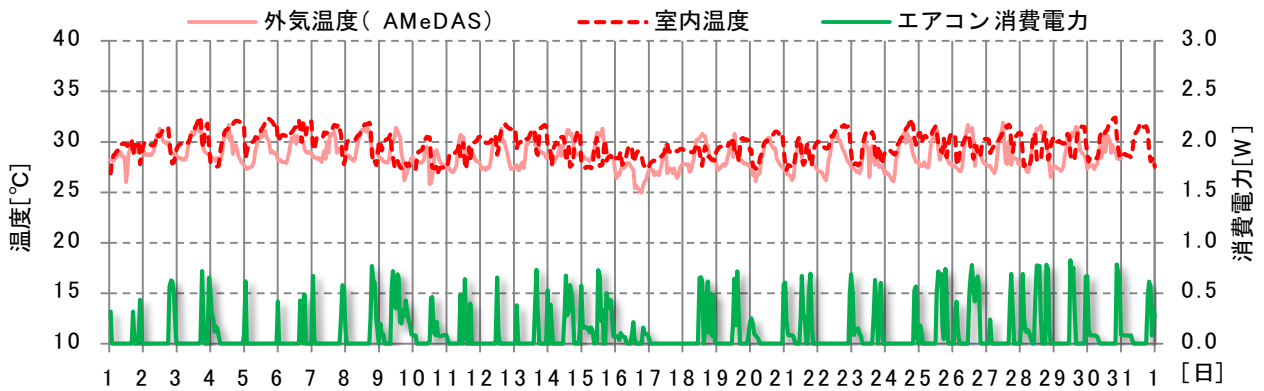


図 28 エアコン稼働状態と室内外温度（2014年8月）

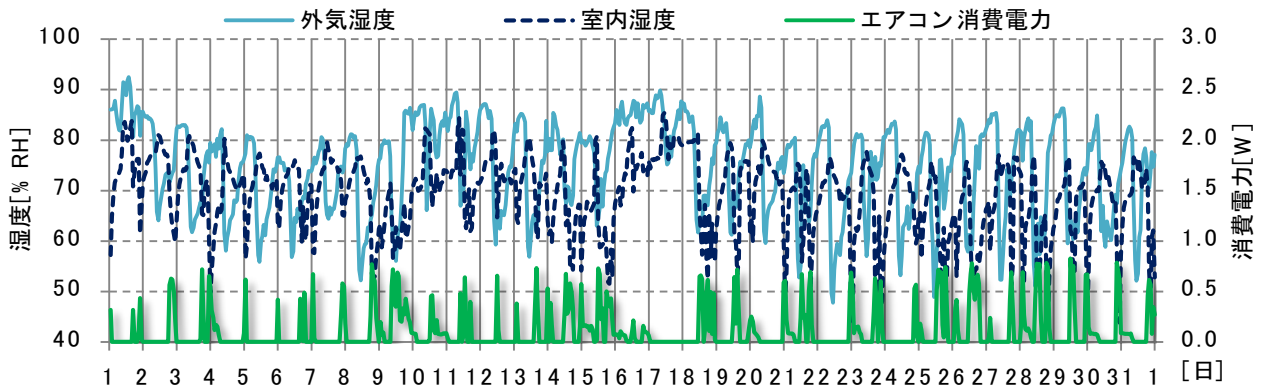


図 29 エアコン稼働状態と室内外湿度（2014年8月）

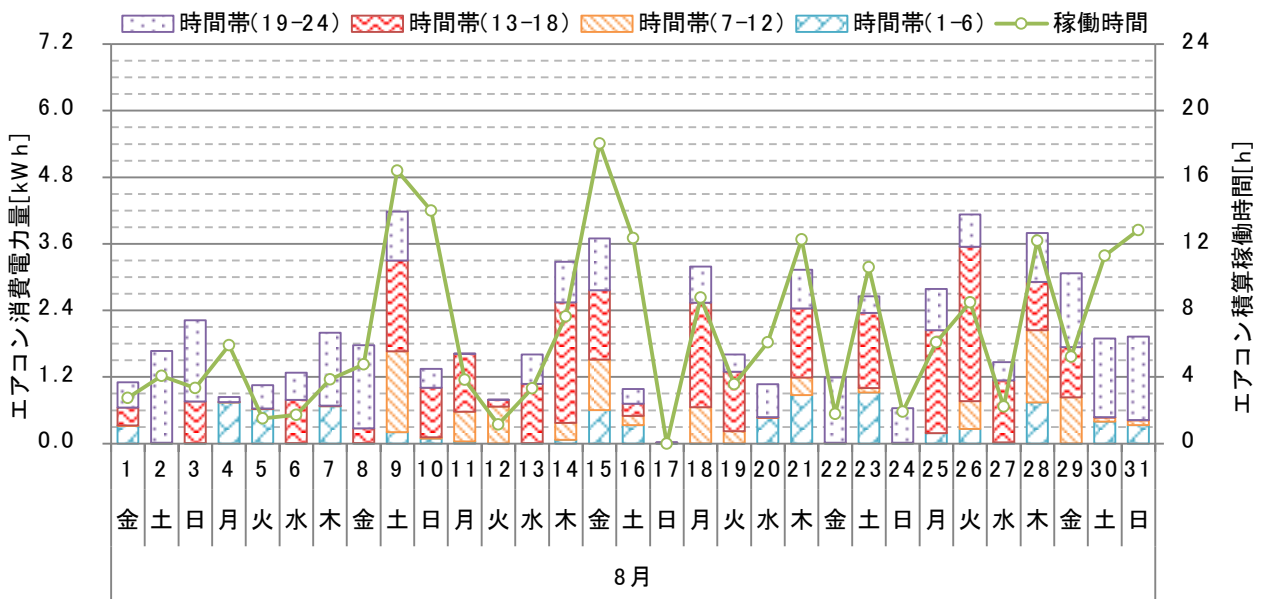


図 30 日別の時間帯別消費電力量と積算稼働時間（2014年8月）

(6) 9月

外気温センサーが鼠害によって故障したため AMeDAS で代用した。

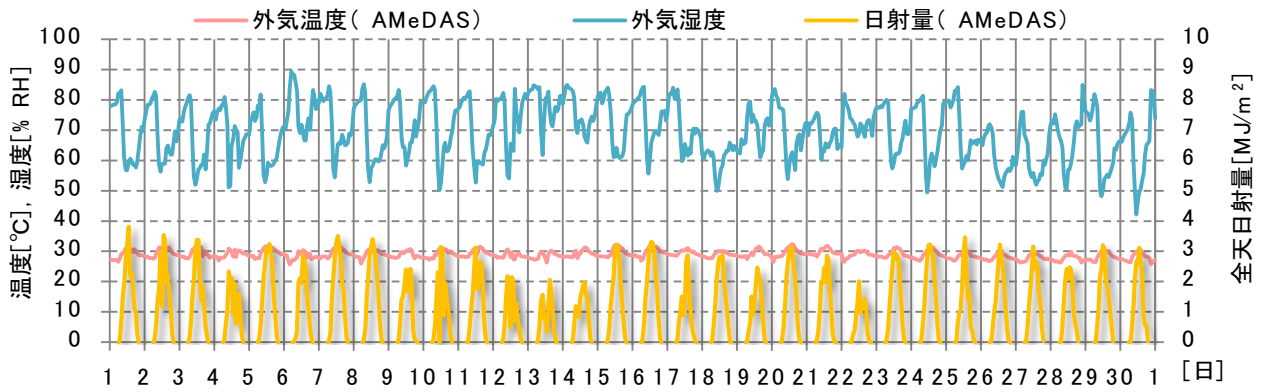


図 31 外気条件 (2014 年 9 月)

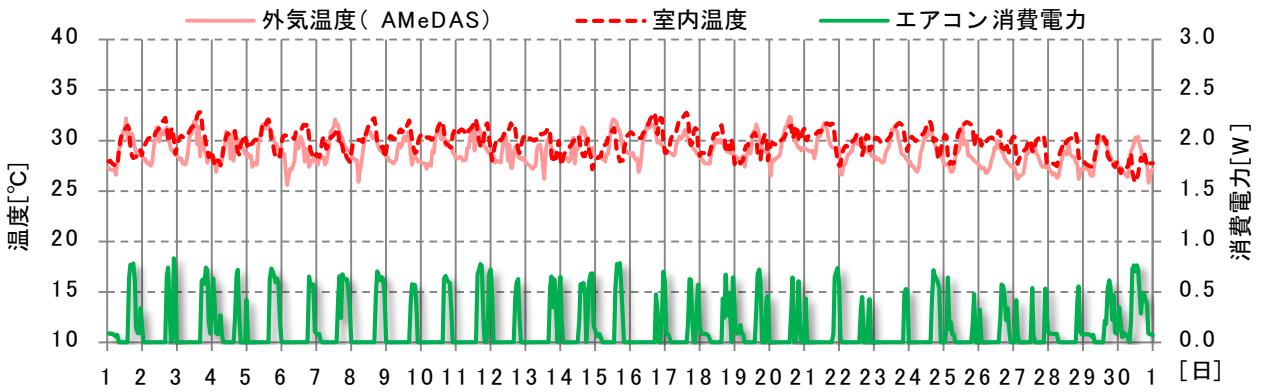


図 32 エアコン稼働状態と室内外温度 (2014 年 9 月)

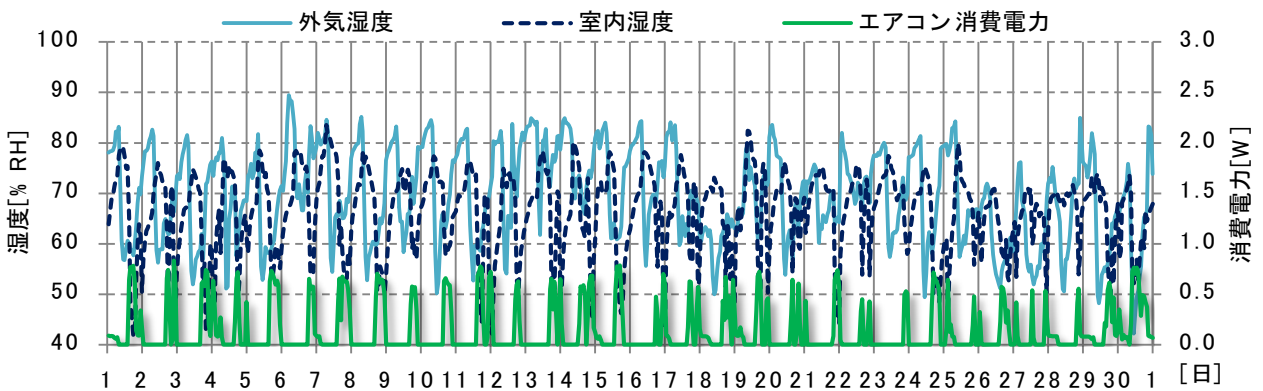


図 33 エアコン稼働状態と室内外湿度 (2014 年 9 月)

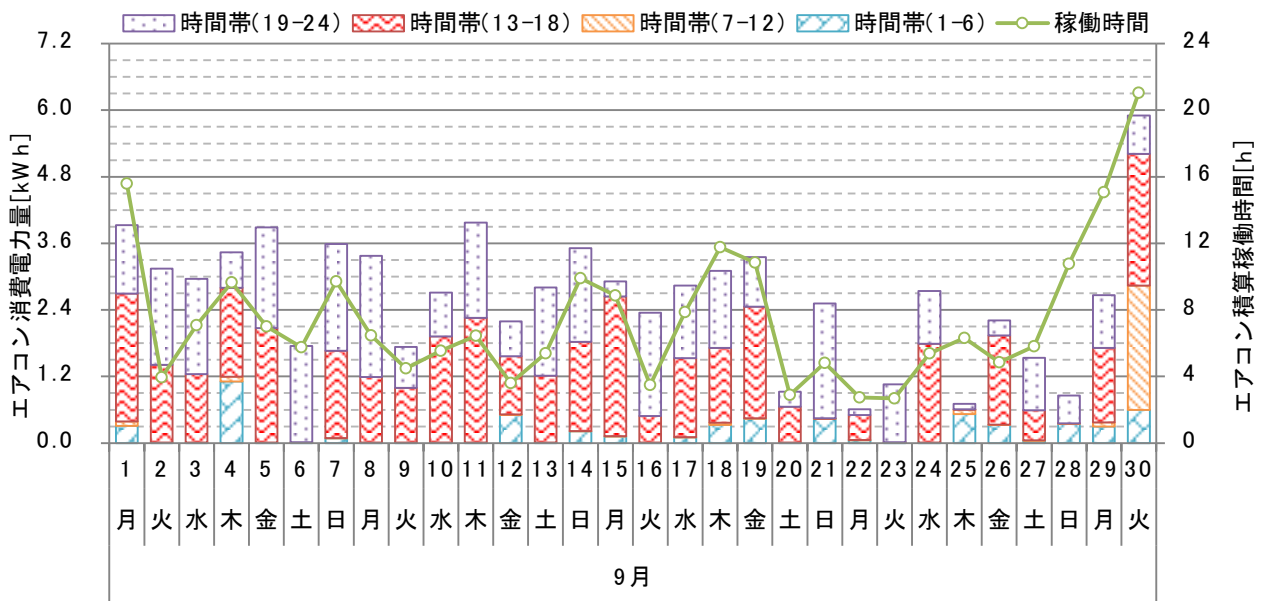


図 34 日別の時間帯別消費電力量と積算稼働時間（2014年9月）

(7) 10月

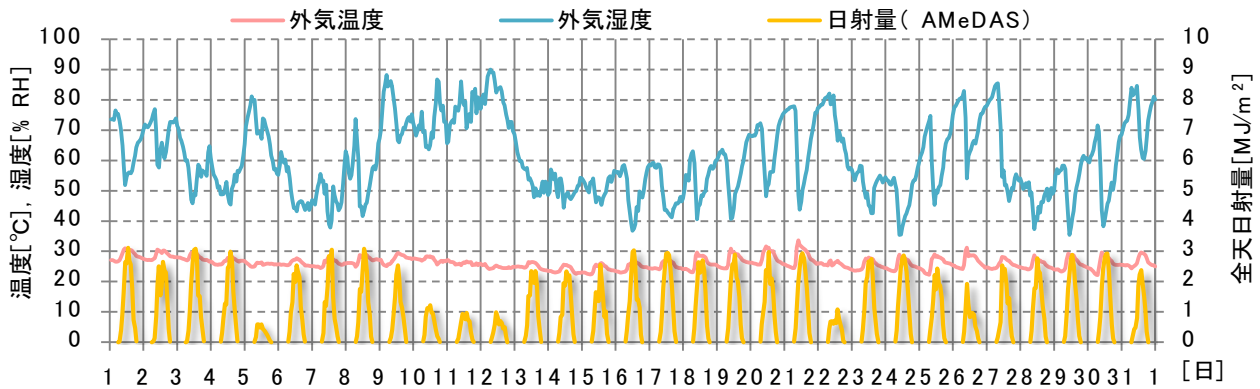


図 35 外気条件（2014年10月）

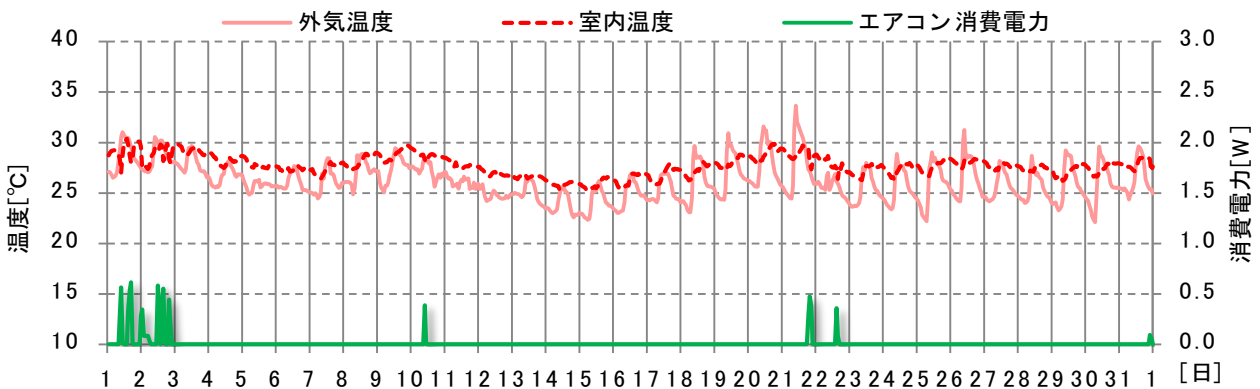


図 36 エアコン稼働状態と室内外温度（2014年10月）

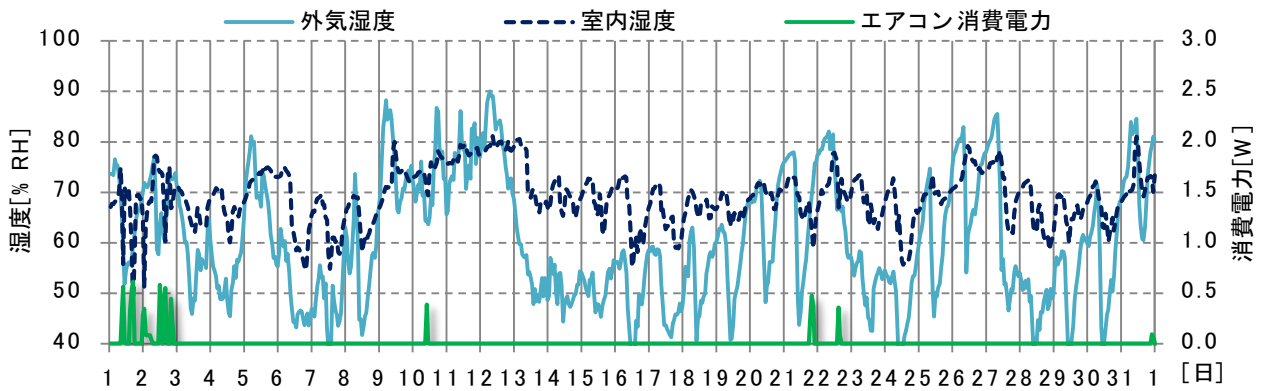


図 37 エアコン稼働状態と室内外湿度 (2014 年 10 月)

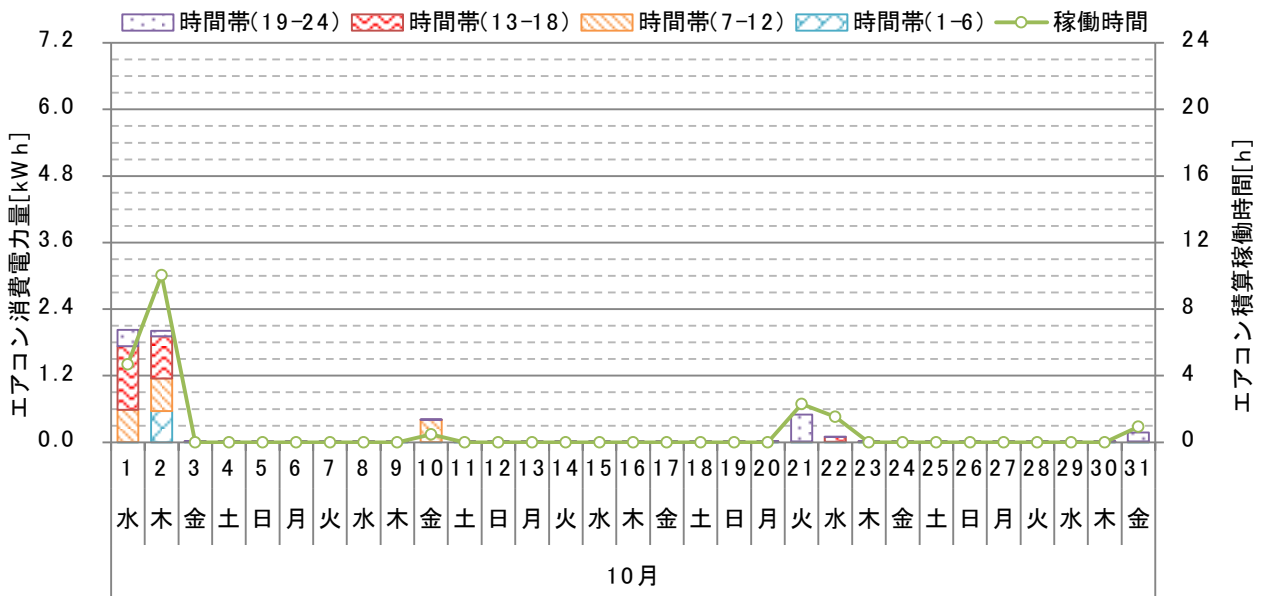


図 38 日別の時間帯別消費電力量と積算稼働時間 (2014 年 10 月)

(8) 11月

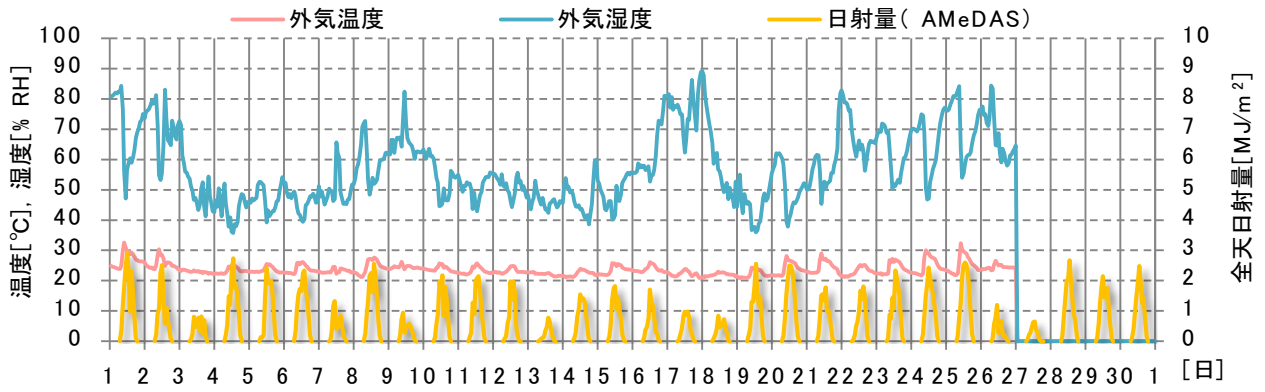


図 39 外気条件 (2014年11月)

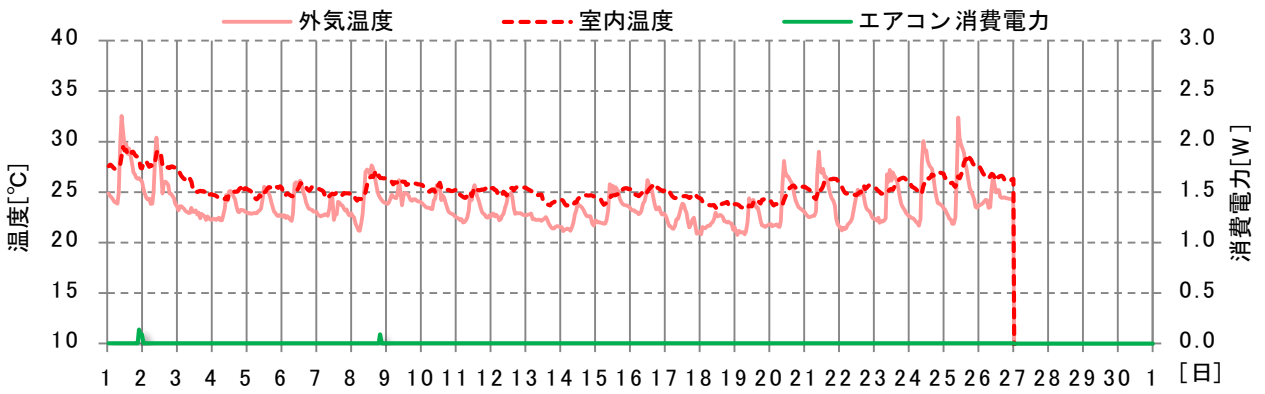


図 40 エアコン稼働状態と室内外温度 (2014年11月)

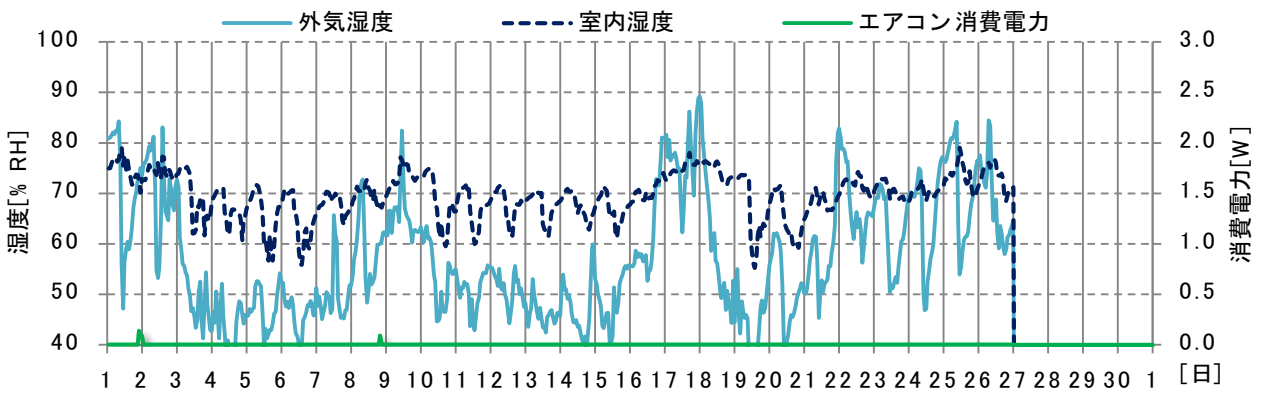


図 41 エアコン稼働状態と室内外湿度 (2014年11月)

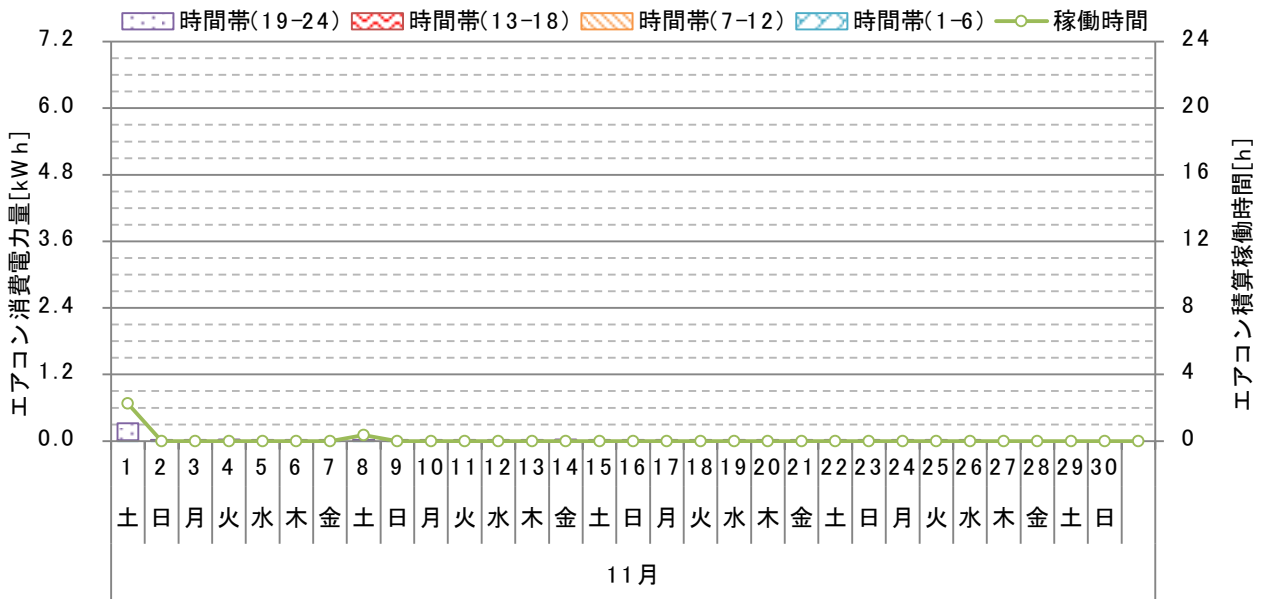


図 42 日別の時間帯別消費電力量と積算稼働時間 (2014 年 11 月)

1.3.2 集合住宅

1.3.2.1 月別使用実態

月別使用実態を時系列グラフで以下に示す。集合住宅では測定期間を通してエアコンを自動制御モードで使用しており、基本的に 24 時間稼働させていた。

(1) 4 月

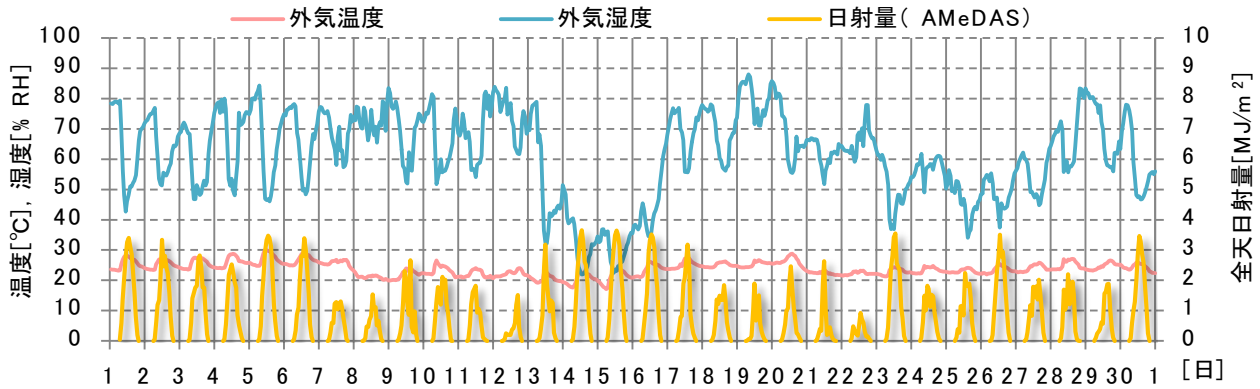


図 43 外気条件 (2015 年 4 月)

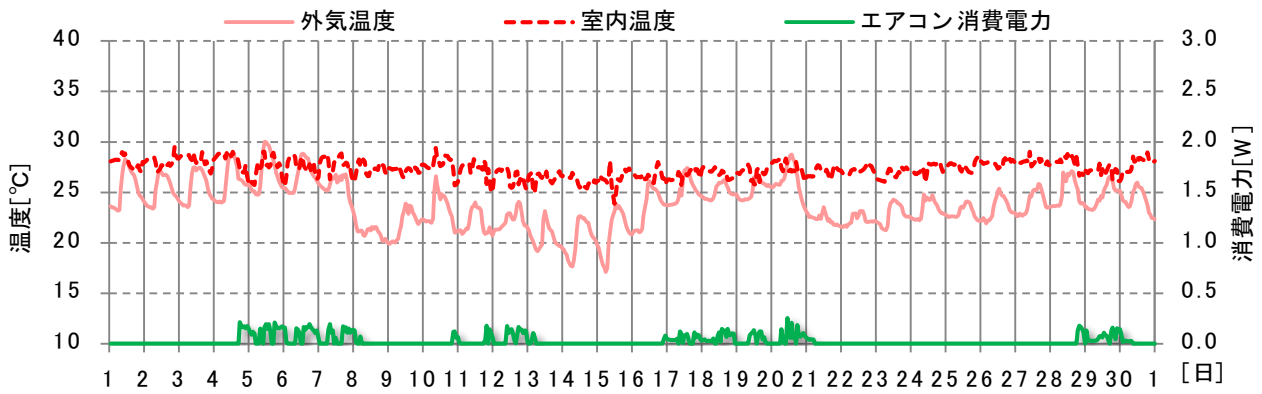


図 44 エアコン稼働状態と室内外温度（2015年4月）

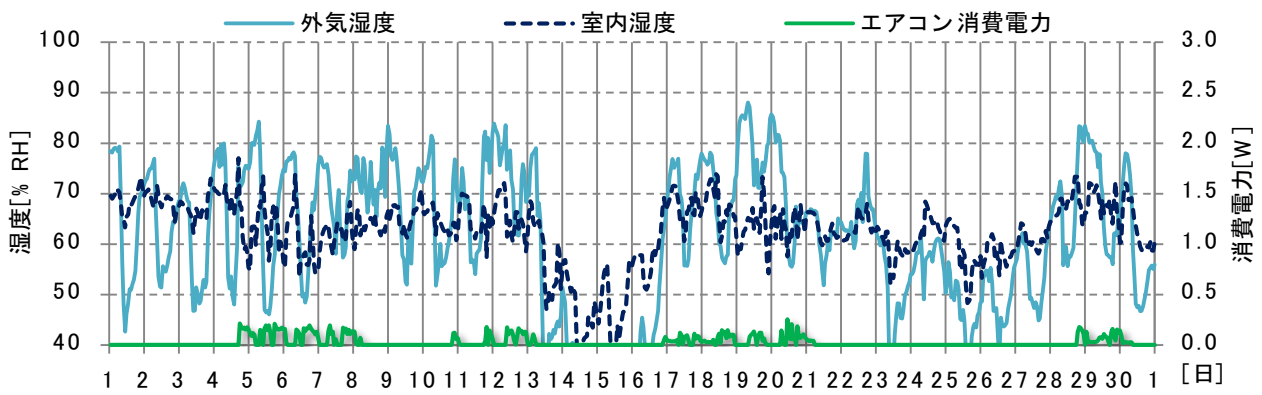


図 45 エアコン稼働状態と室内外湿度（2015年4月）

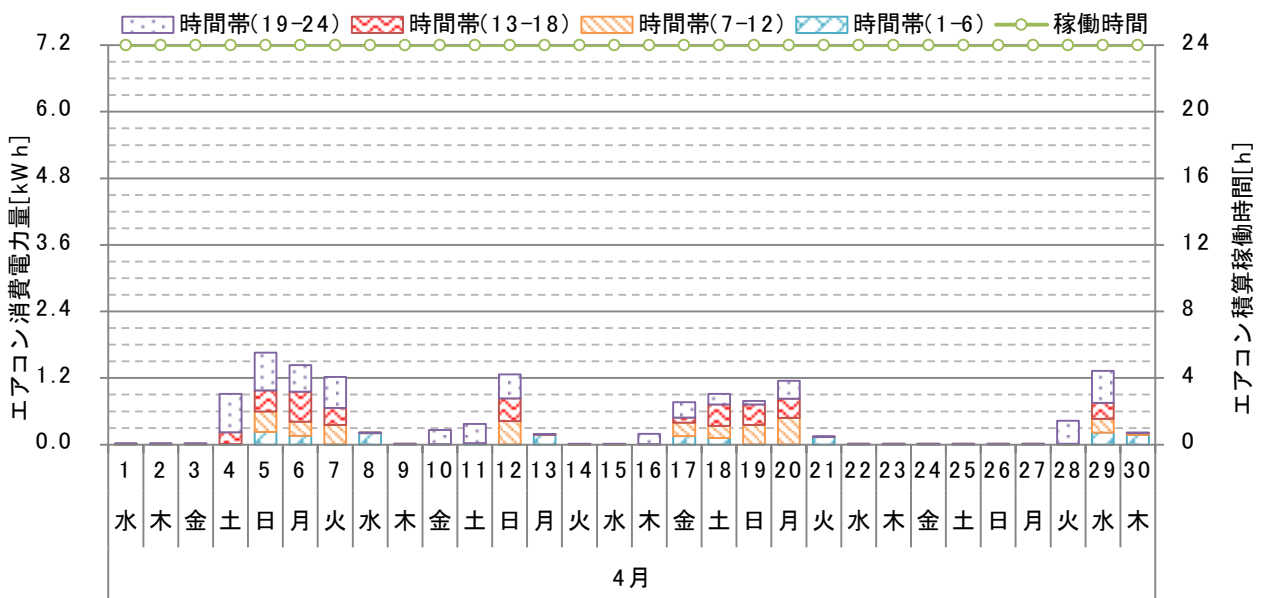


図 46 日別の時間帯別消費電力量と積算稼働時間（2015年4月）

(2) 5月

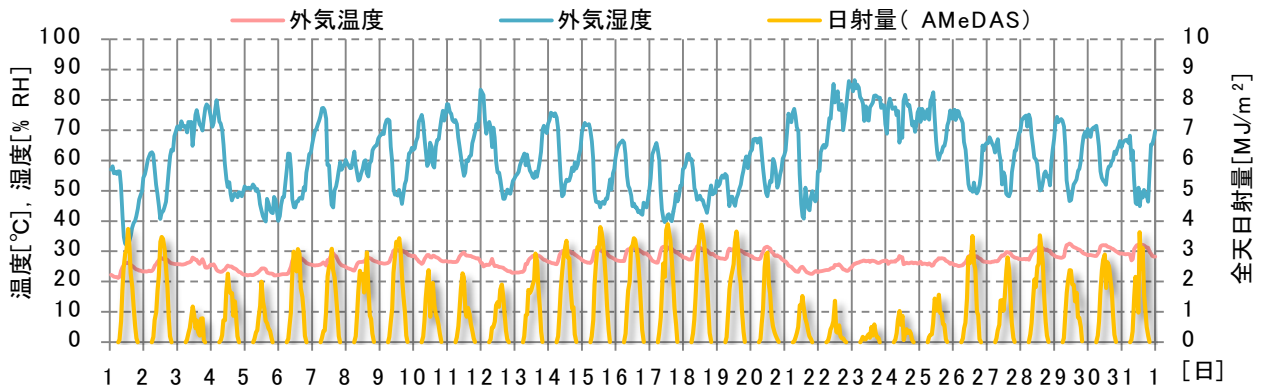


図 47 外気条件 (2015年5月)

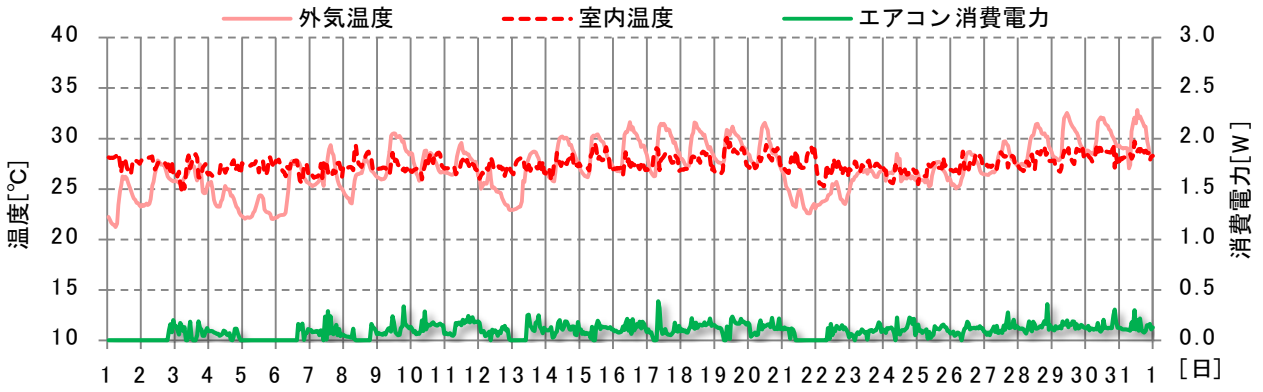


図 48 エアコン稼働状態と室内外温度 (2015年5月)

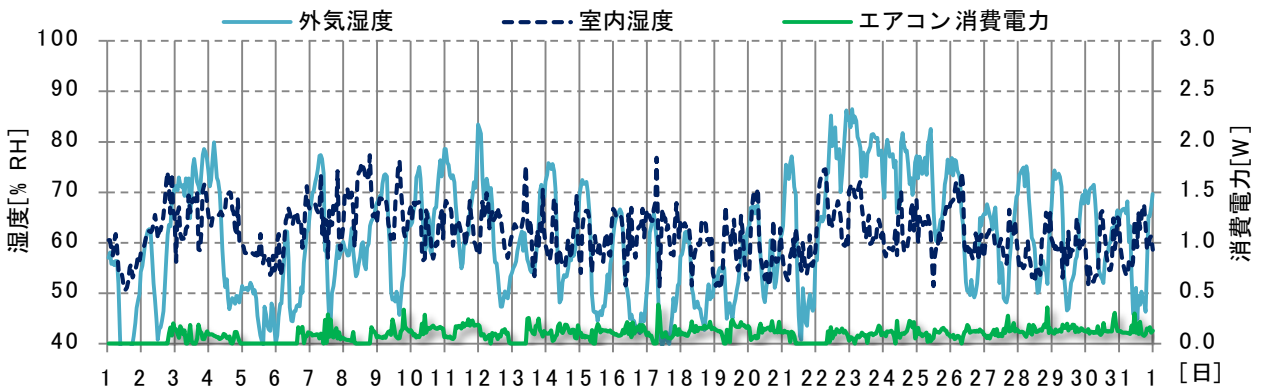


図 49 エアコン稼働状態と室内外湿度 (2015年5月)

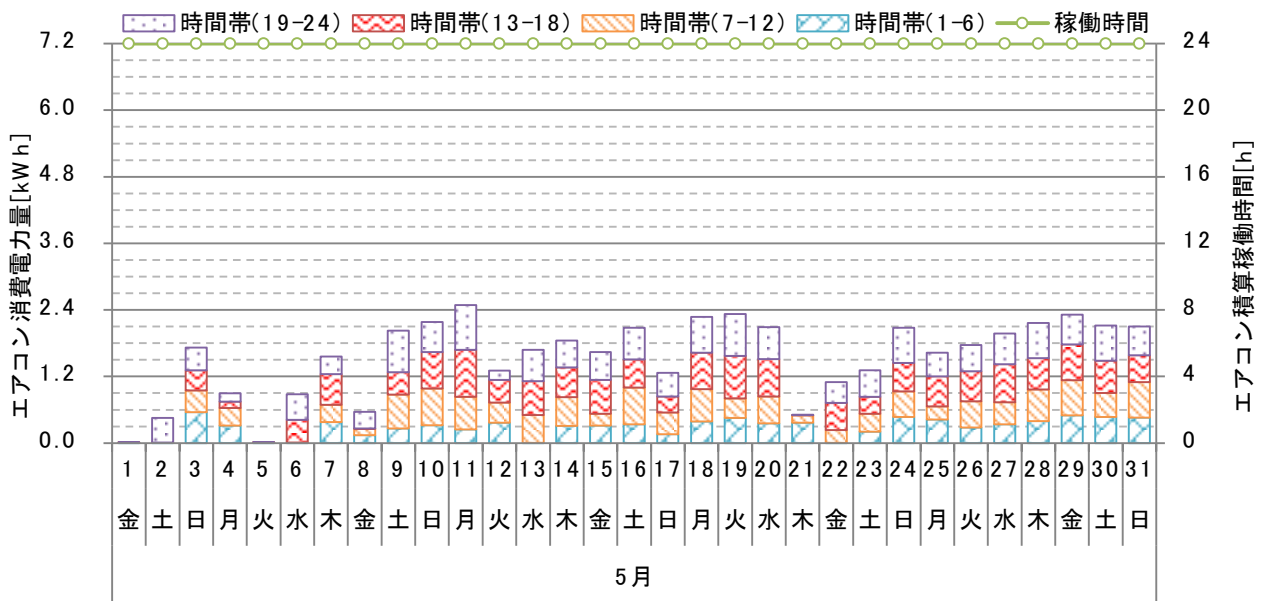


図 50 日別の時間帯別消費電力量と積算稼働時間（2015年5月）

(3) 6月

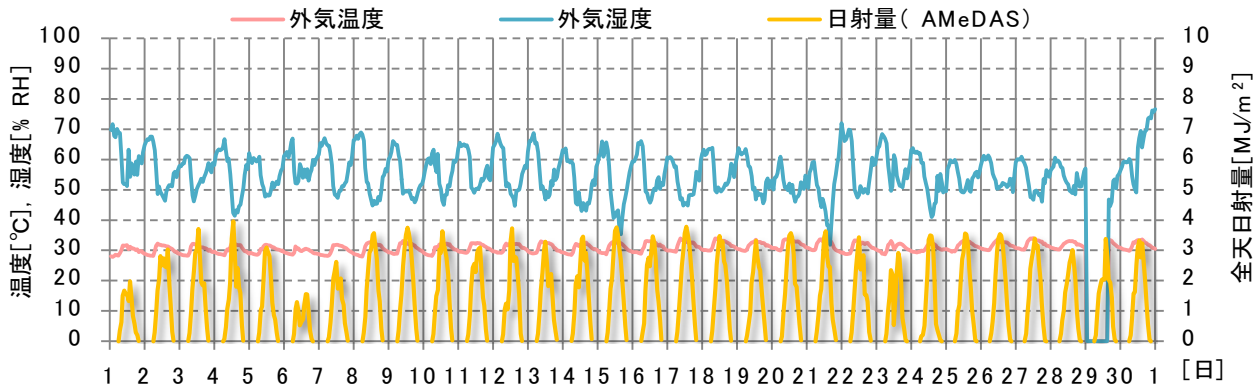


図 51 外気条件（2015年6月）

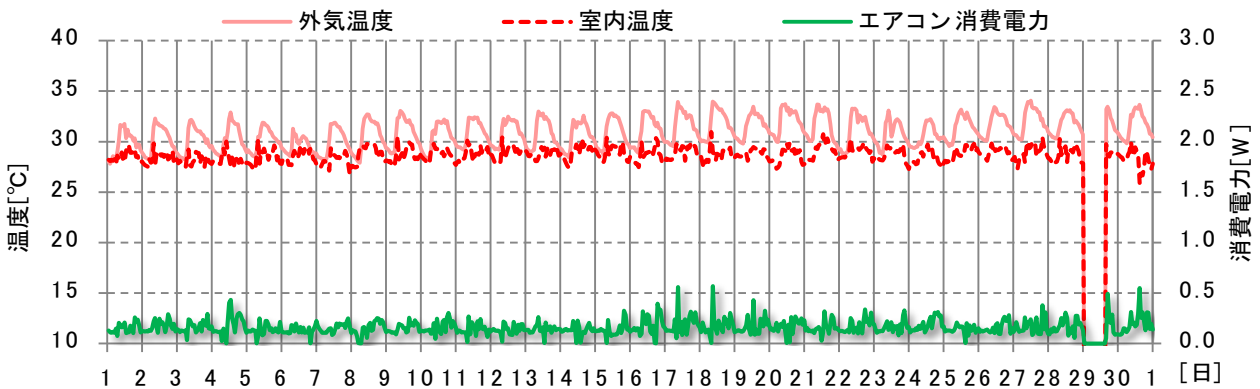


図 52 エアコン稼働状態と室内外温度（2015年6月）

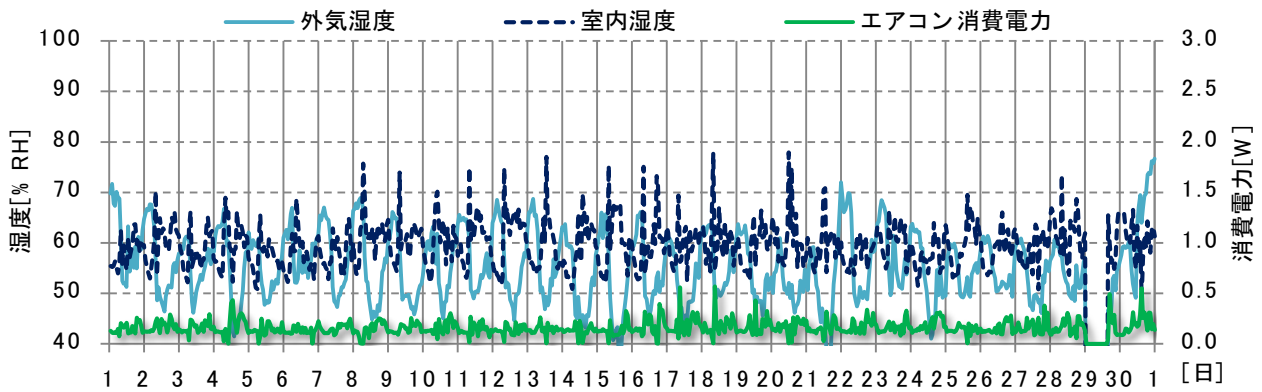


図 53 エアコン稼働状態と室内外湿度 (2015年6月)

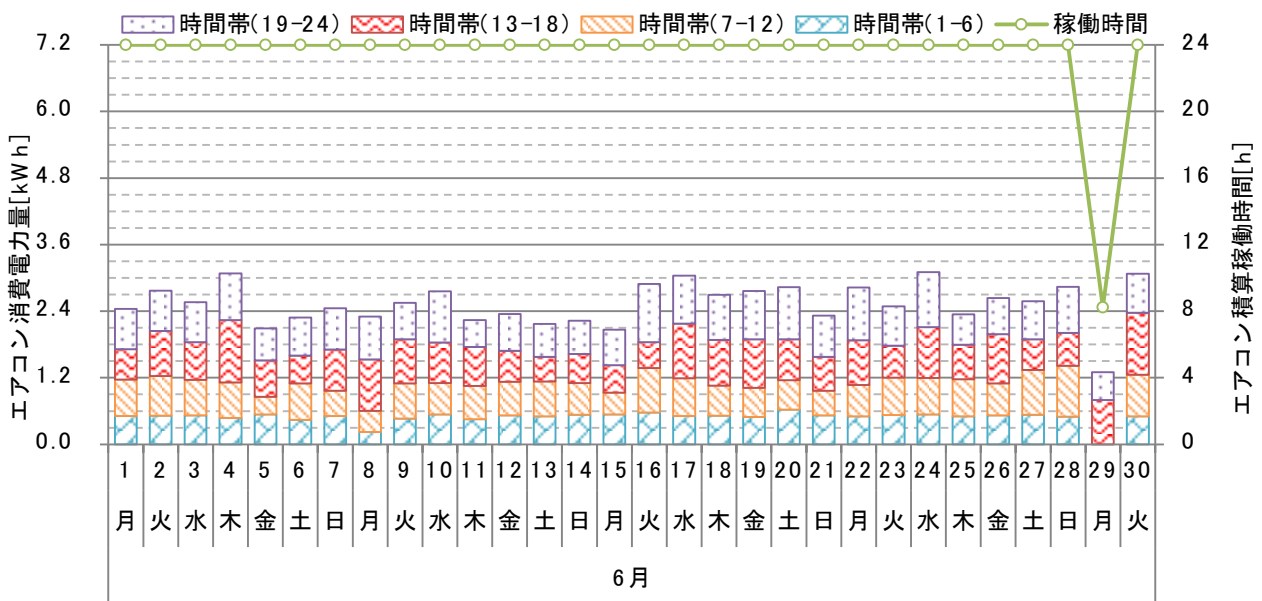


図 54 日別の時間帯別消費電力量と積算稼働時間 (2015年6月)

(4) 7月

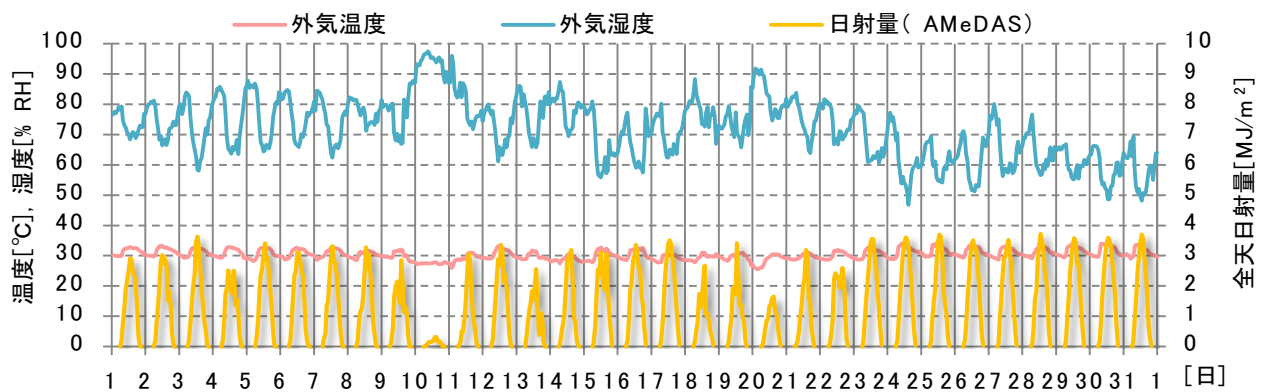


図 55 外気条件 (2015年7月)

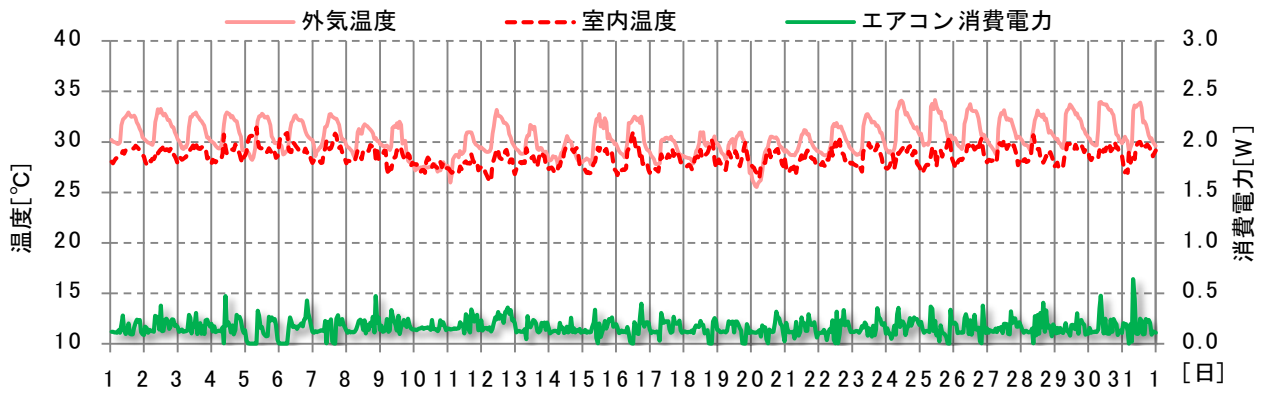


図 56 エアコン稼働状態と室内外温度 (2015年7月)

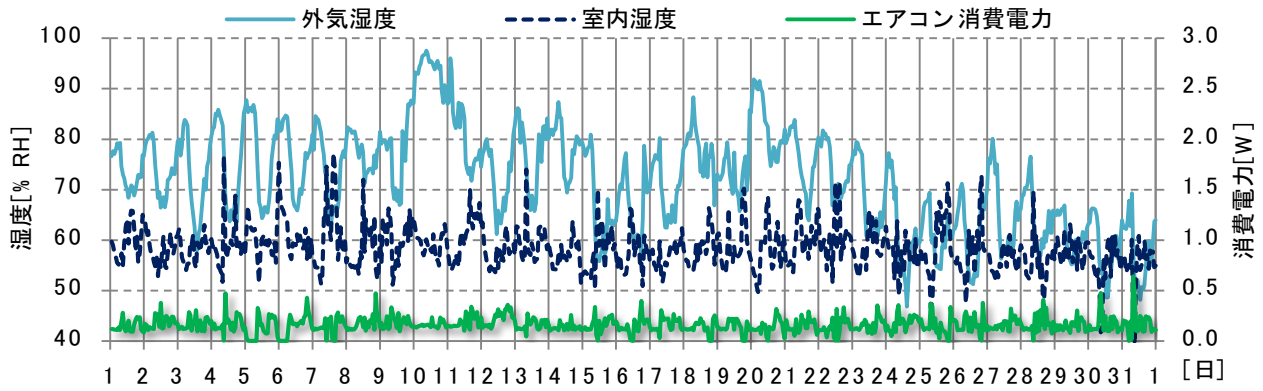


図 57 エアコン稼働状態と室内外湿度 (2015年7月)

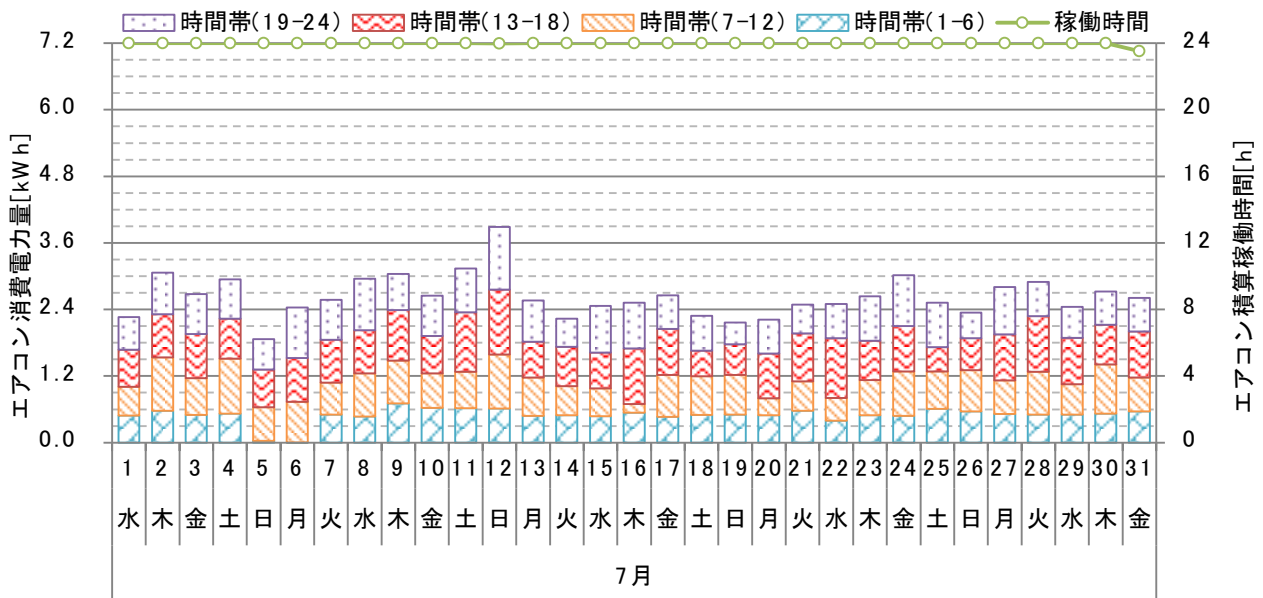


図 58 日別の時間帯別消費電力量と積算稼働時間 (2015年7月)

(5) 8月

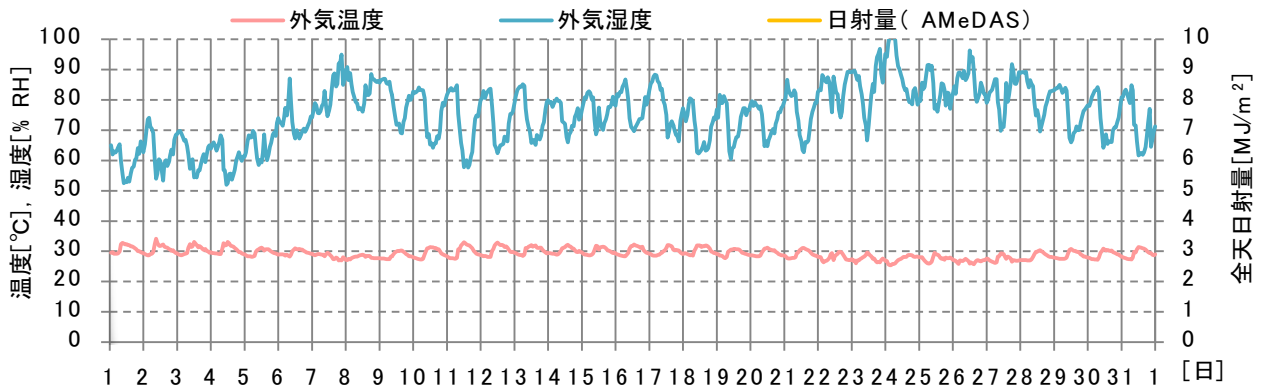


図 59 外気条件 (2015 年 8 月)

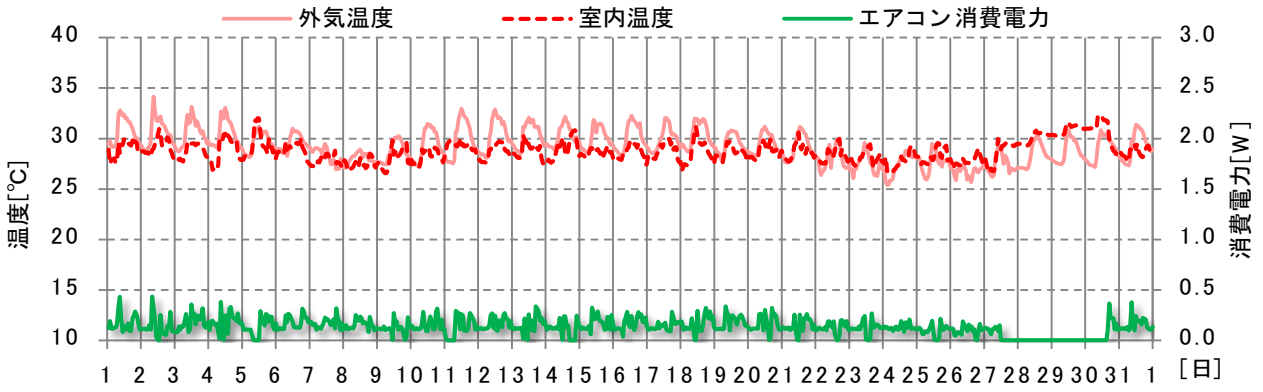


図 60 エアコン稼働状態と室内外温度 (2015 年 8 月)

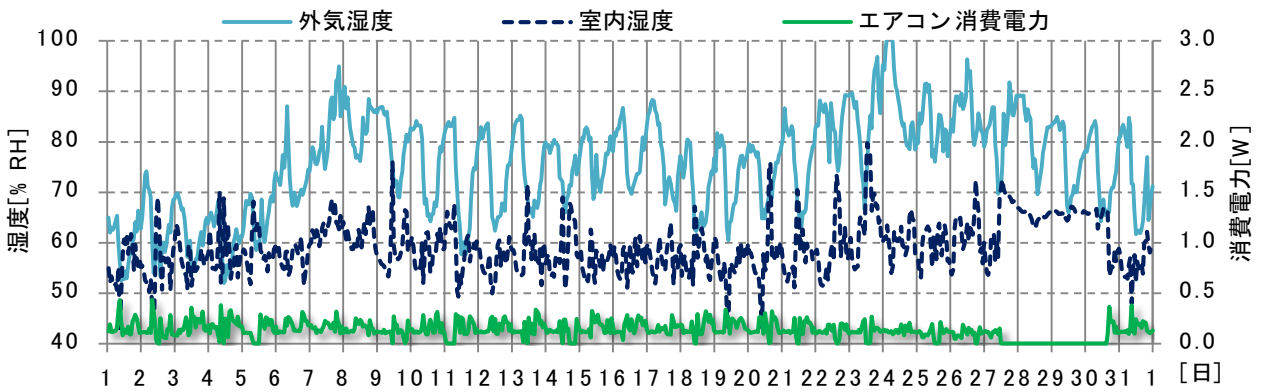


図 61 エアコン稼働状態と室内外湿度 (2015 年 8 月)

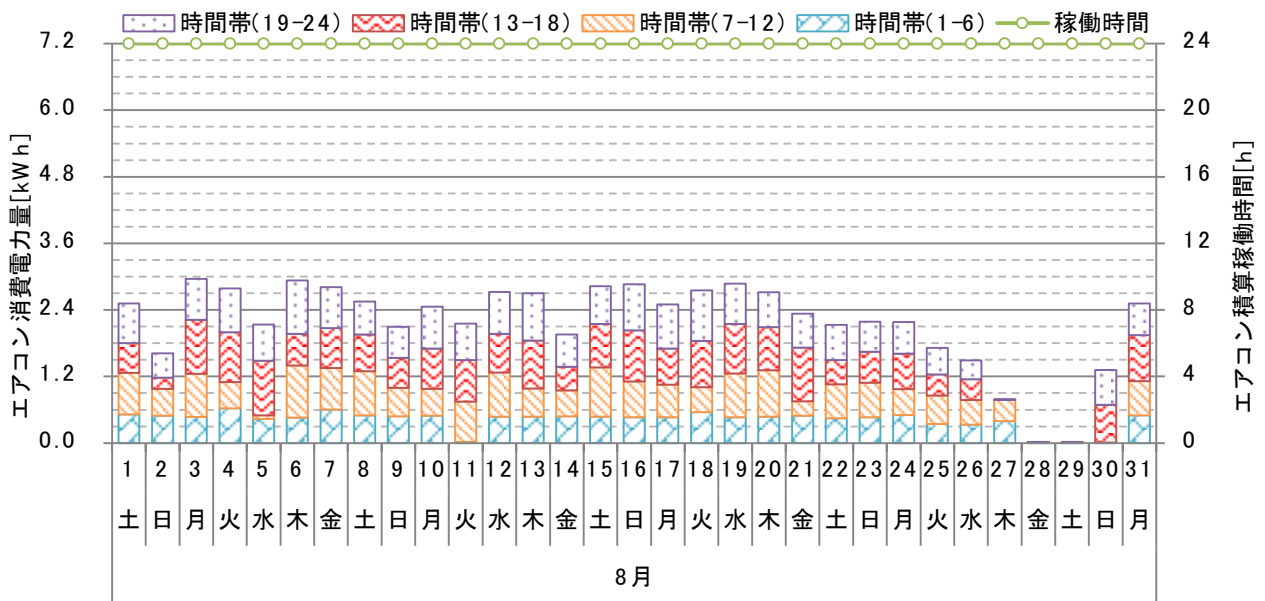


図 62 日別の時間帯別消費電力量と積算稼働時間（2015年8月）

(6) 9月

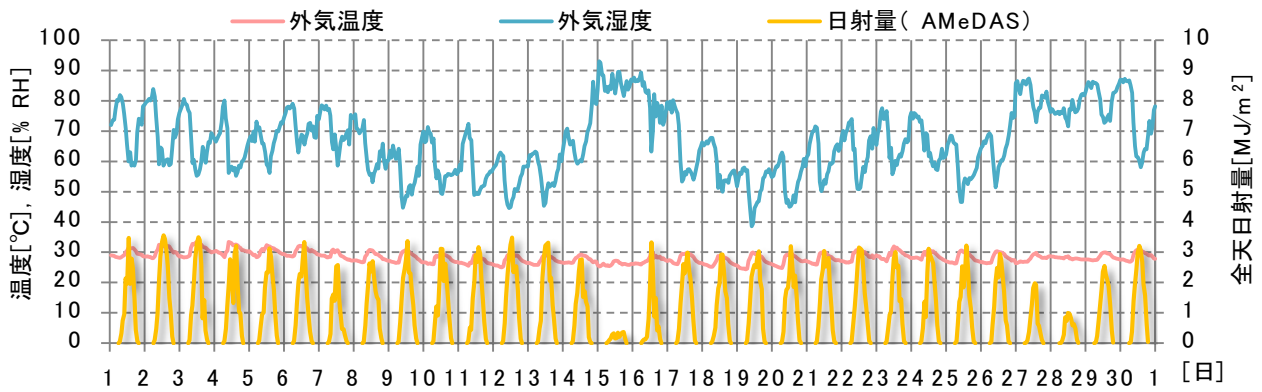


図 63 外気条件（2015年9月）

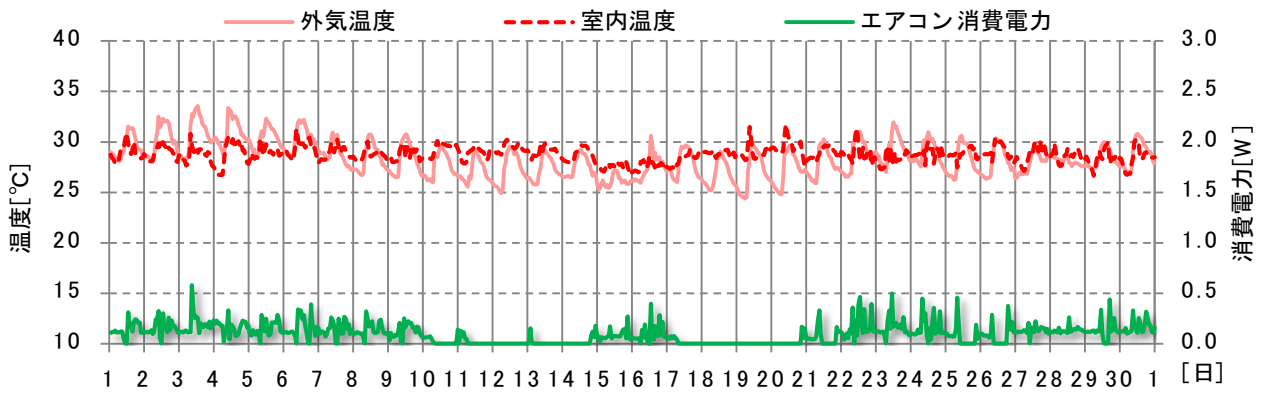


図 64 エアコン稼働状態と室内外温度 (2015 年 9 月)



図 65 エアコン稼働状態と室内外湿度 (2015 年 9 月)

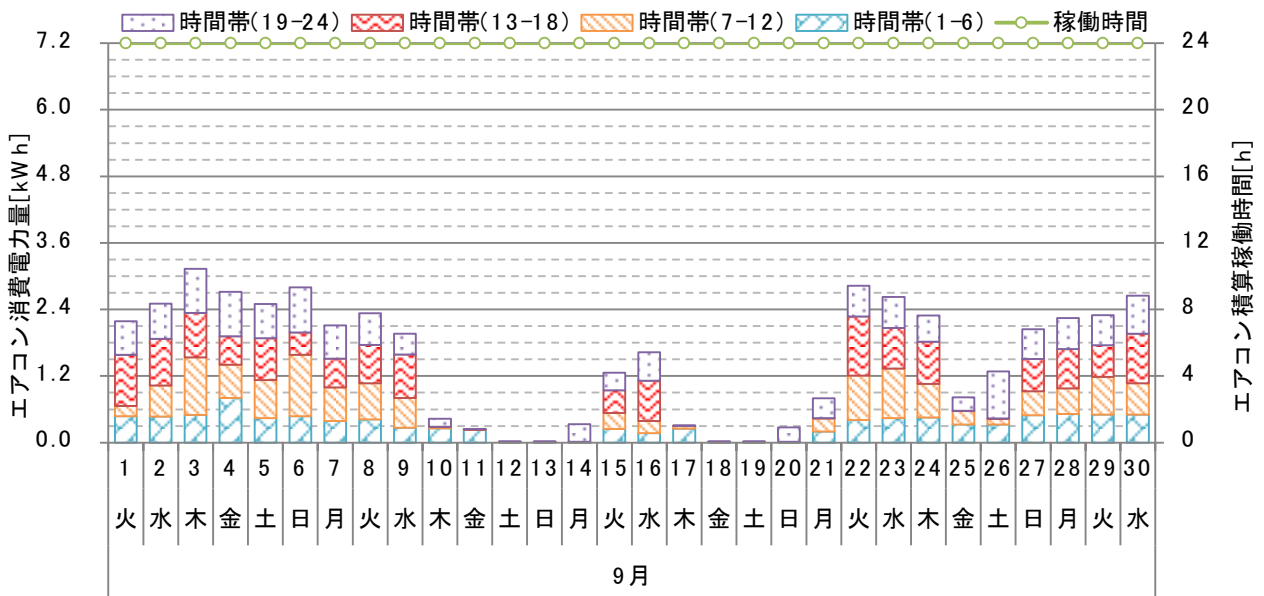


図 66 日別の時間帯別消費電力量と積算稼働時間 (2015 年 9 月)

(7) 10月

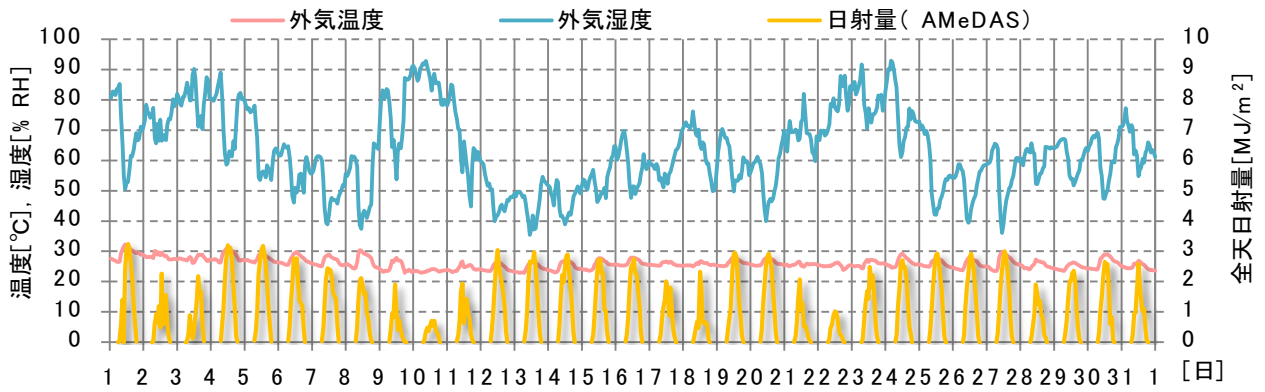


図 67 外気条件 (2015年10月)

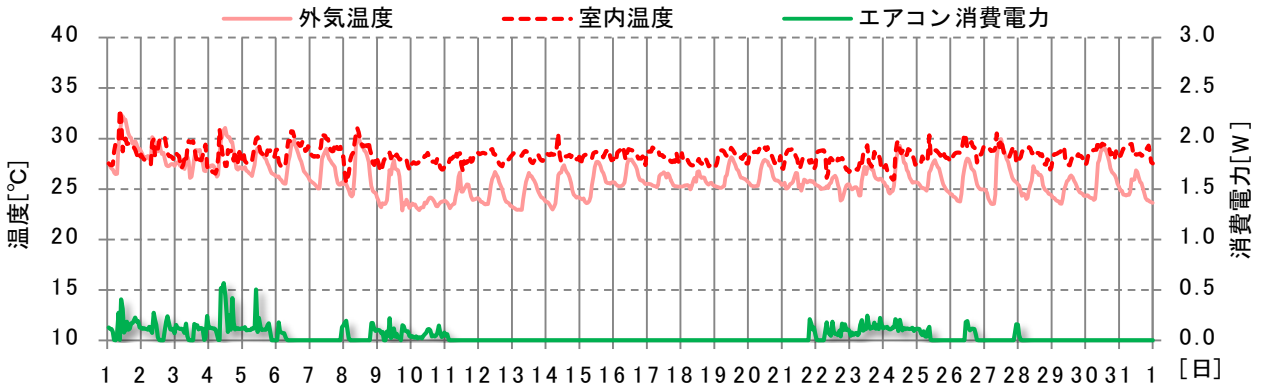


図 68 エアコン稼働状態と室内外温度 (2015年10月)

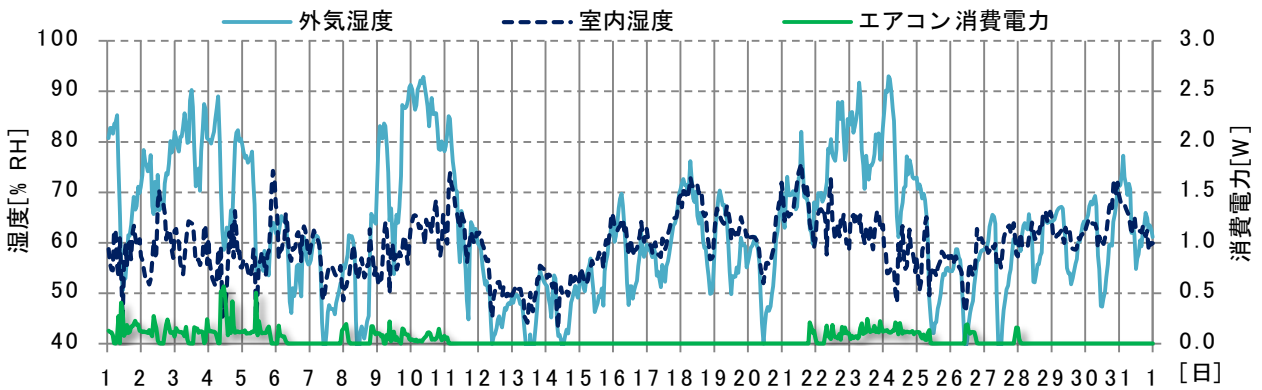


図 69 エアコン稼働状態と室内外湿度 (2015年10月)

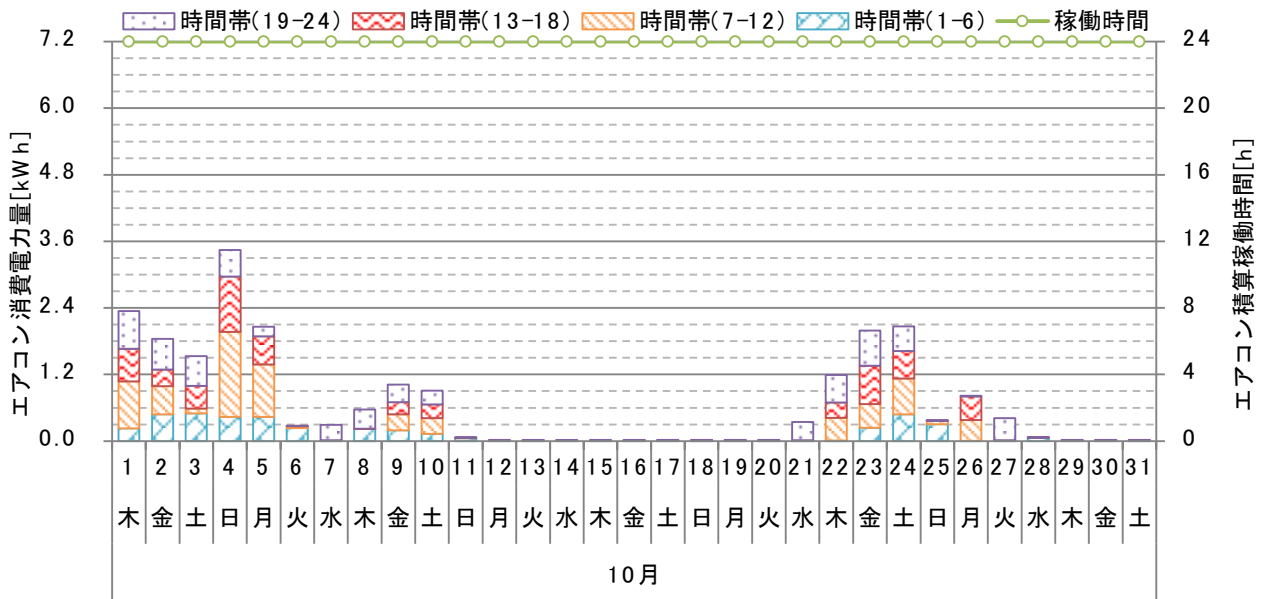


図 70 日別の時間帯別消費電力量と積算稼働時間 (2015 年 10 月)

(8) 11 月

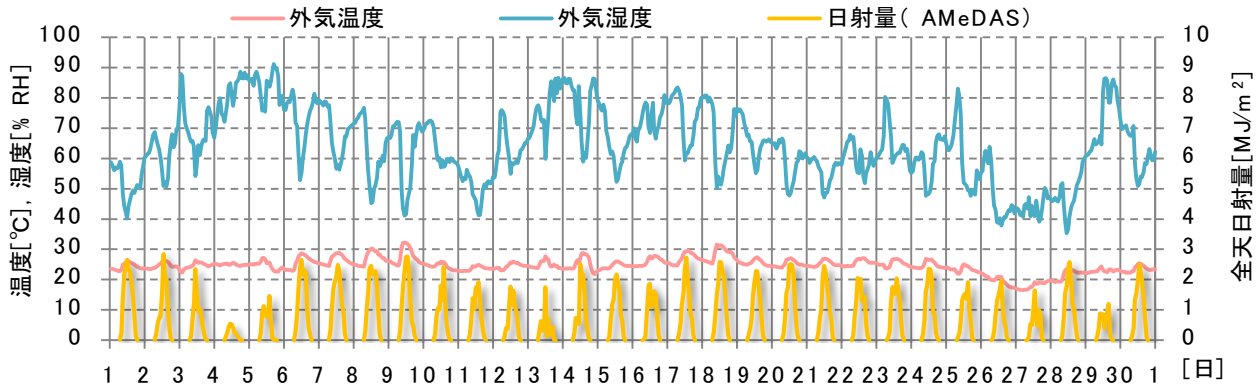


図 71 外気条件 (2015 年 11 月)

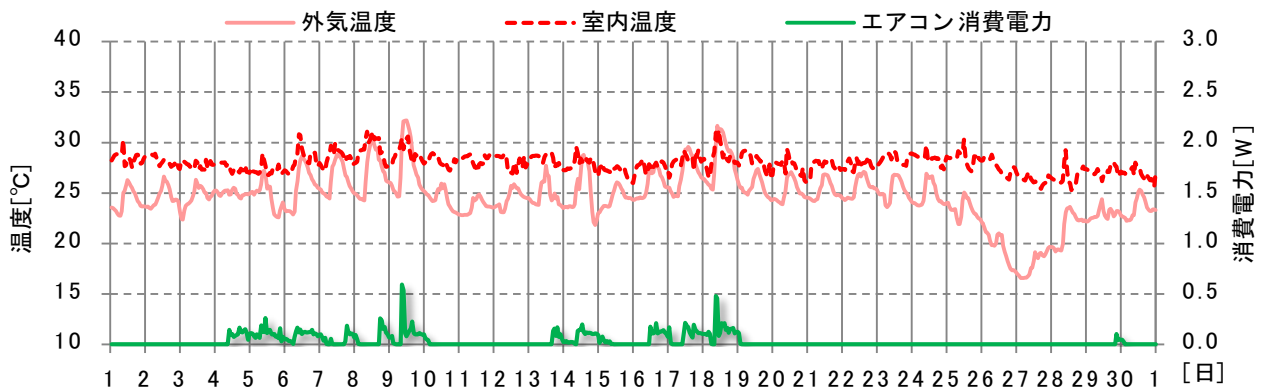


図 72 エアコン稼働状態と室内外温度 (2015 年 11 月)

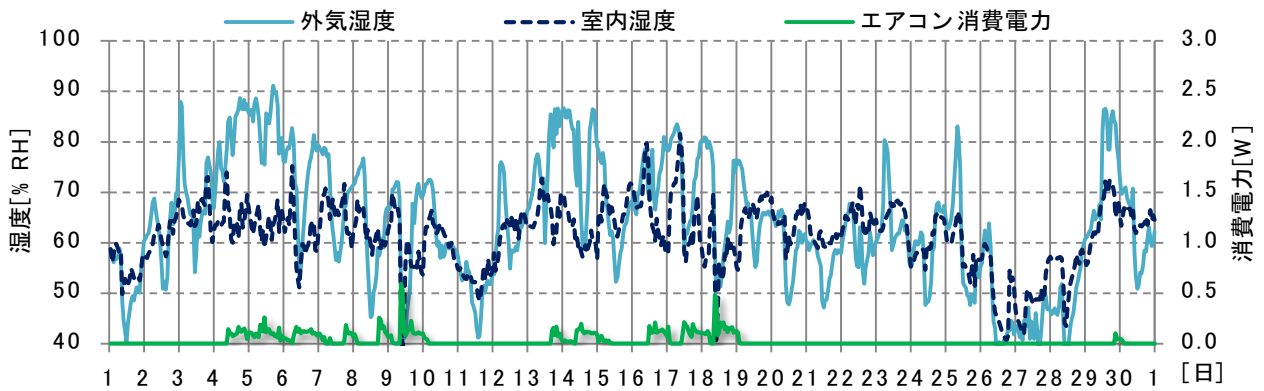


図 73 エアコン稼働状態と室内外湿度 (2015 年 11 月)

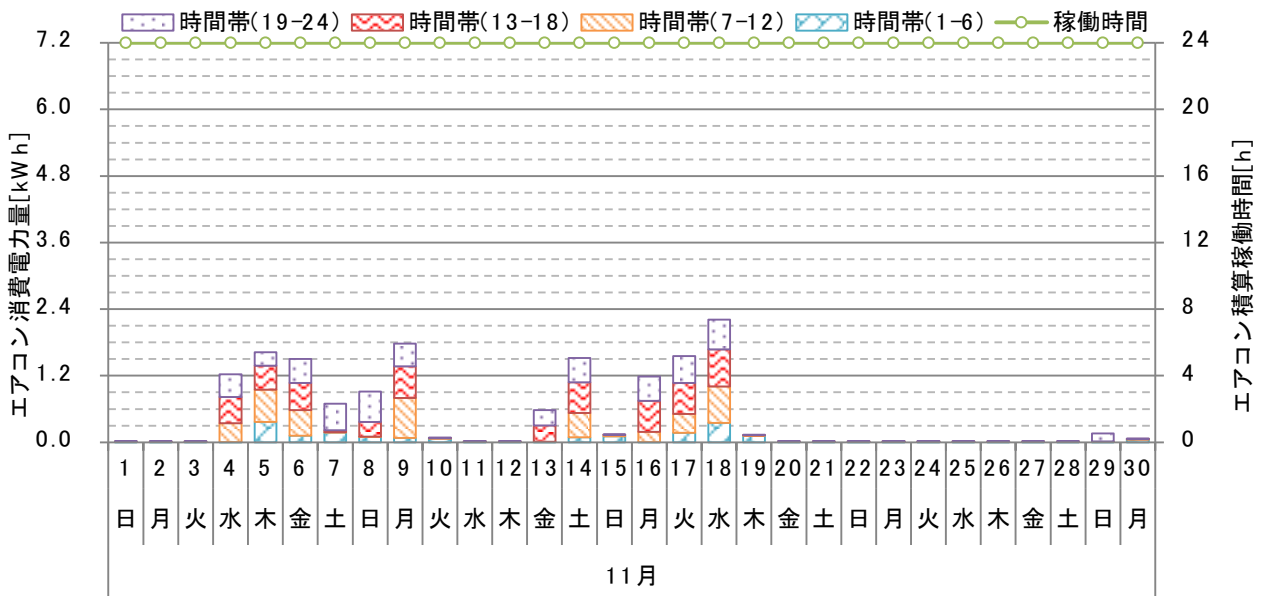


図 74 日別の時間帯別消費電力量と積算稼働時間 (2015 年 11 月)

1.3.2.2 制御固定試験

特定の 1 週間においてエアコン制御が温熱環境および効率に与える影響を把握するため、運転制御を固定した試験を行った。居住者は、1 週間「冷房」もしくは「除湿」制御のみでエアコンを発停させ居住空間の温熱環境を調整した。

(1) 冷房

運転制御を冷房に固定し、2015/9/14~9/20 の 1 週間使用した結果を以下に示す。外気温度が 25~32℃という低い温度帯で推移し、居室は 5 人の在室者や家電などの内部発熱がある事から、冷房時にも室外よりも室内の温度が高い傾向にあった。湿度はエアコン稼働時は外気よりも室内の方が低く抑えられていた。

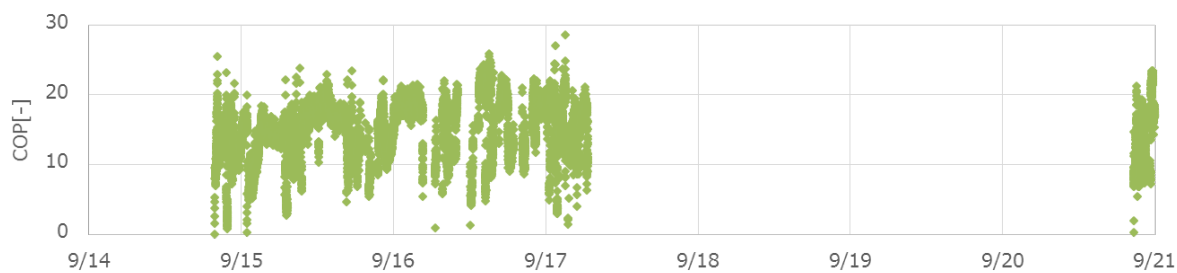


図 75 COP の推移 (冷房)

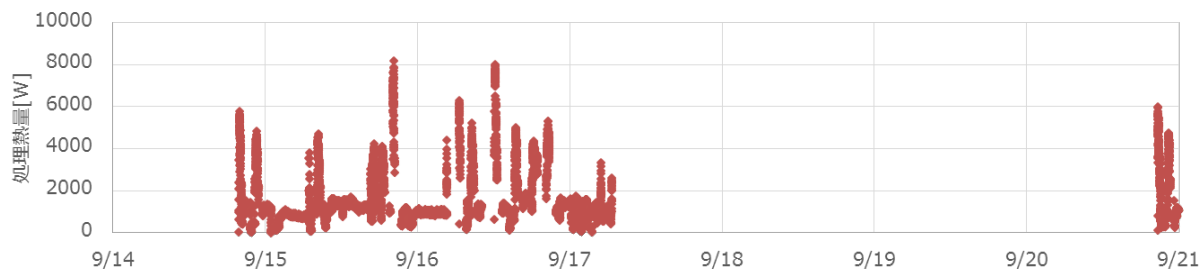


図 76 処理熱量の推移 (冷房)

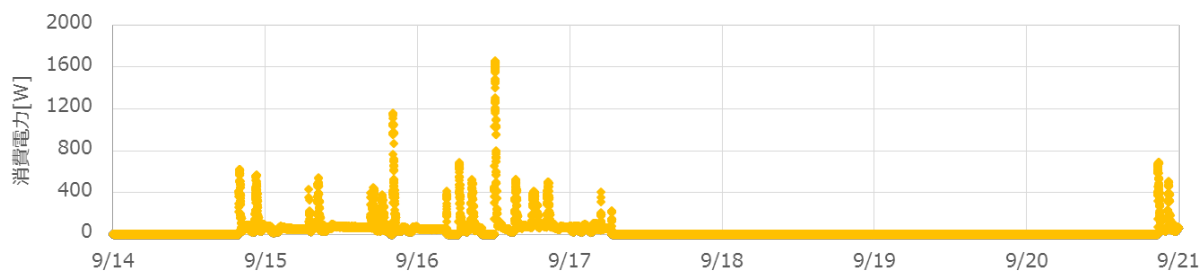


図 77 消費電力の推移 (冷房)

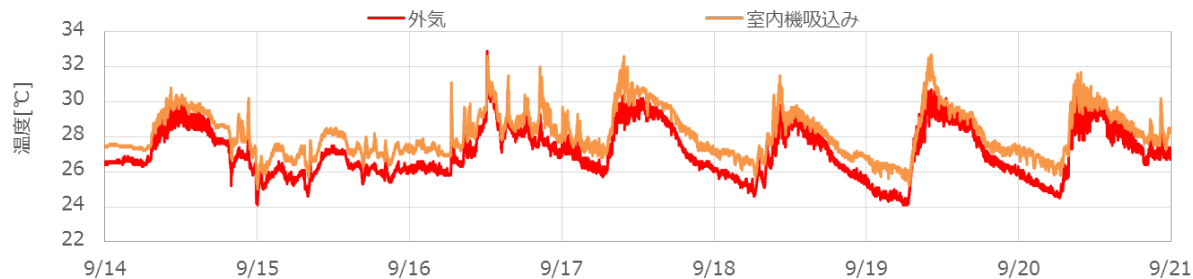


図 78 室内外温度の推移 (冷房)

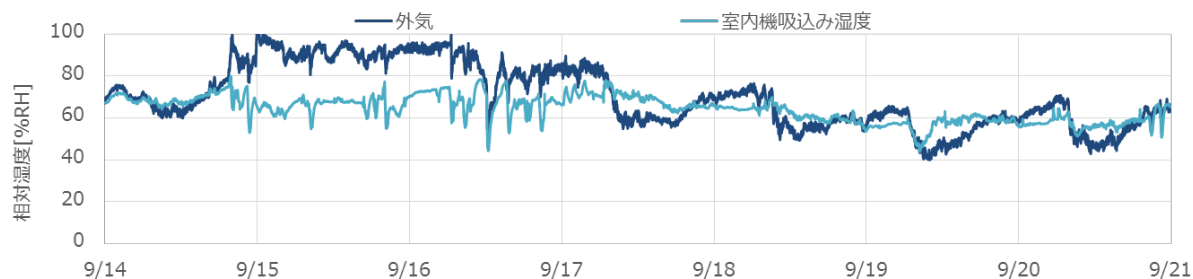


図 79 室内外相対湿度の推移 (冷房)

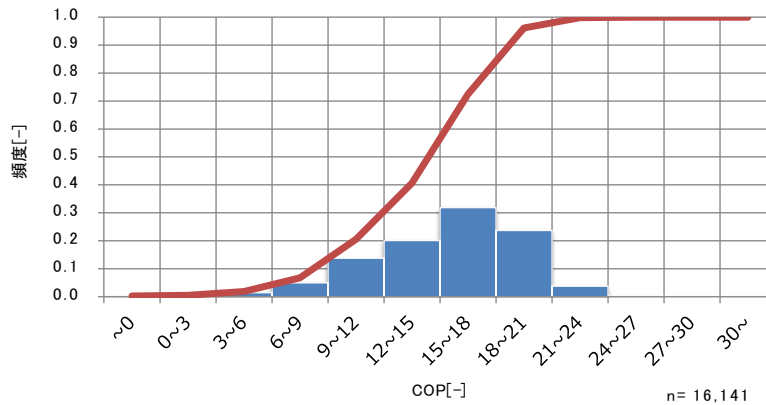


図 80 COP の頻度分布 (冷房)

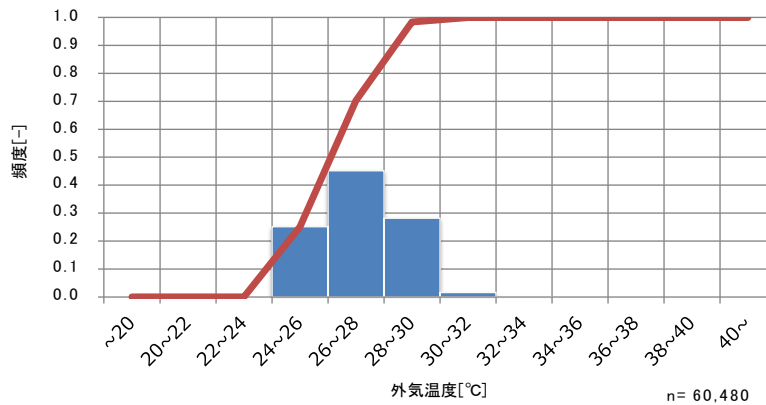


図 81 外気温度の頻度分布 (冷房)

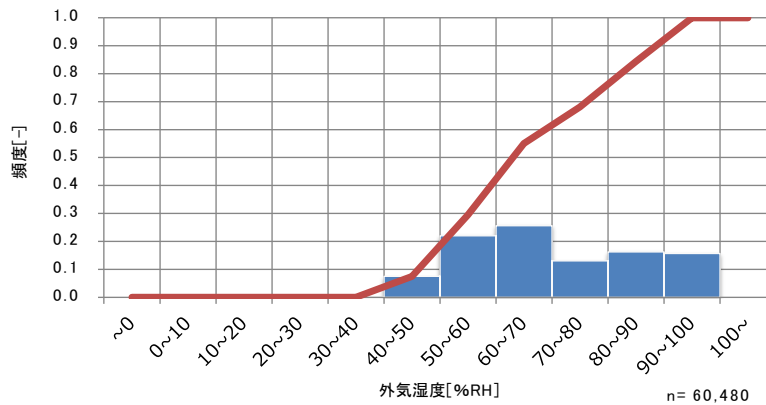


図 82 外気相対湿度の頻度分布 (冷房)

(2) 除湿

運転制御を除湿に固定し、2015/9/21~9/27 の 1 週間使用した結果を以下に示す。外気温度が 26~34 °C であり冷房試験期間よりも高い温度帯で推移した。室内温度は冷房時と同様に室外よりも室内の温度が高い傾向にあった。湿度はエアコン稼働時は外気よりも室内の方が低く抑えられていた。COP は冷房時よりも低い頻度分布となった。これは外気温度の影響もあったと考えられる。

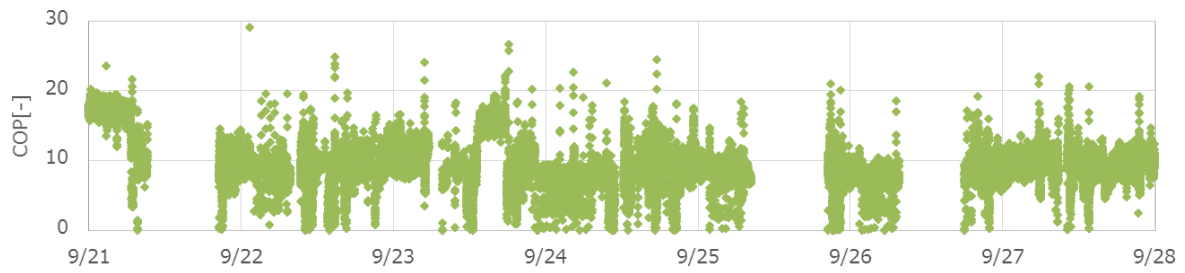


図 83 COP の推移 (除湿)

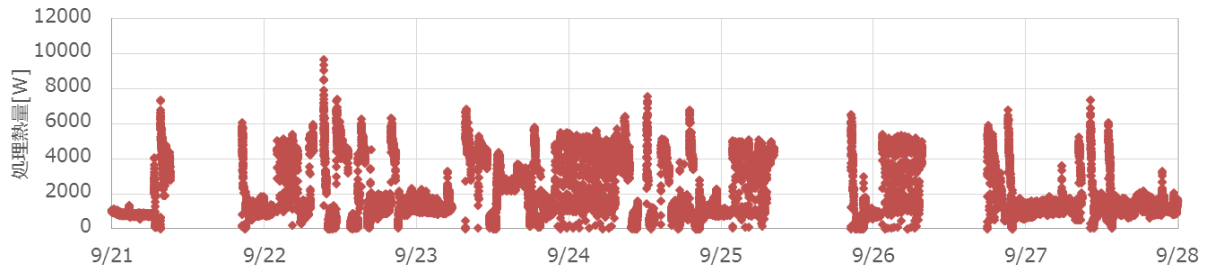


図 84 処理熱量の推移 (除湿)

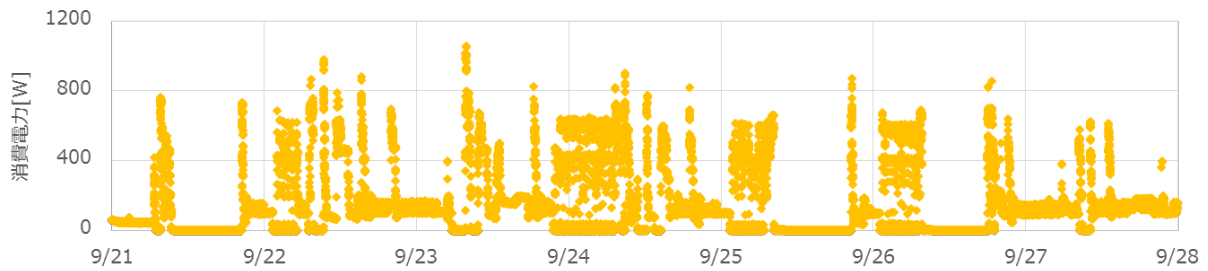


図 85 消費電力の推移 (除湿)

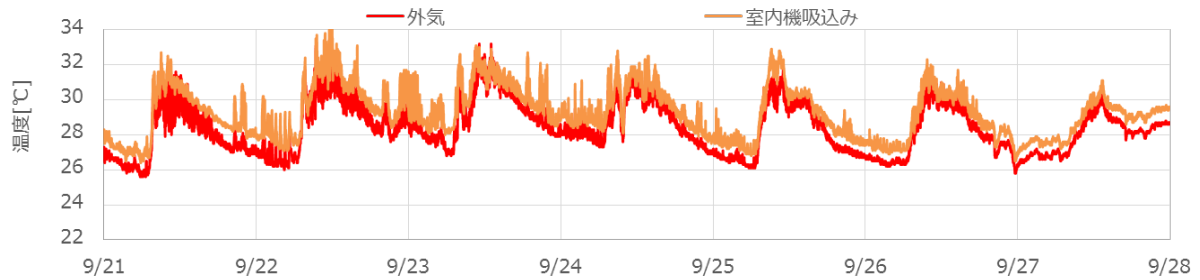


図 86 室内外温度の推移 (除湿)

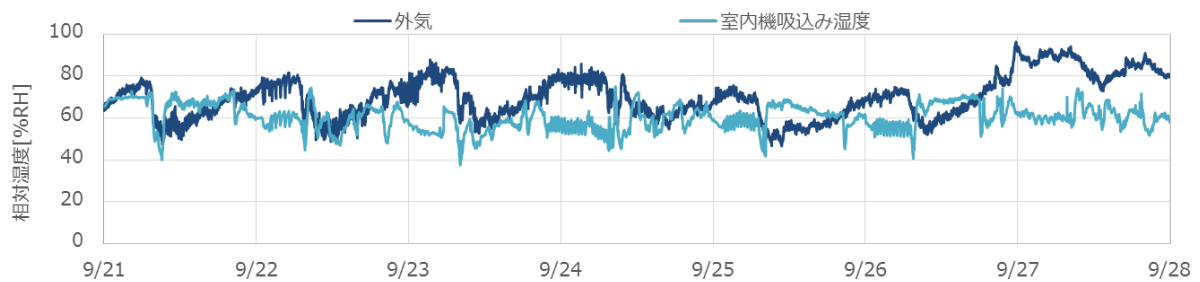


図 87 室内外相対湿度の推移 (除湿)

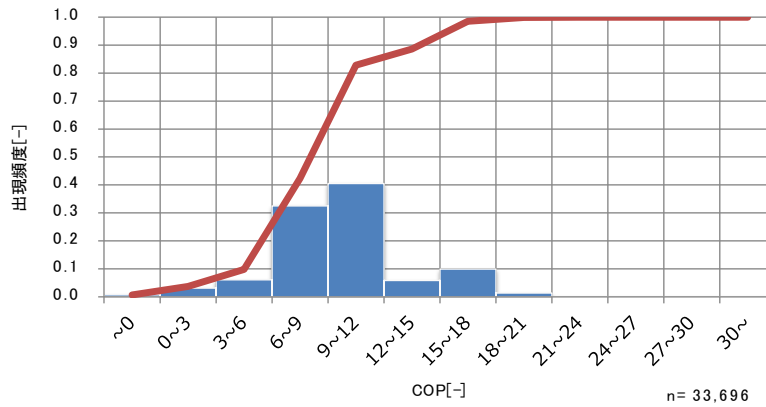


図 88 COP の頻度分布 (除湿)

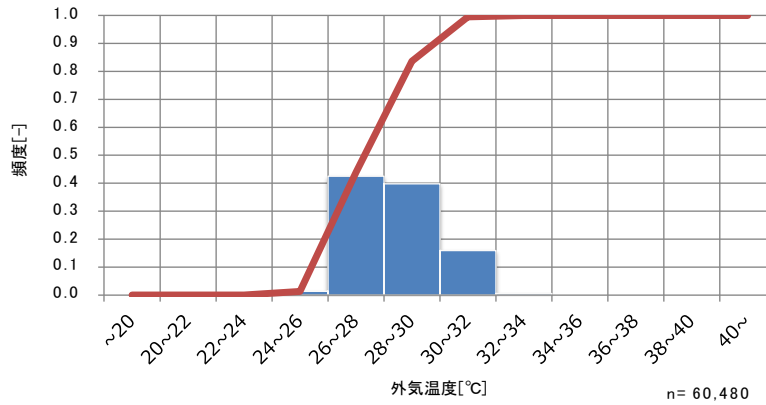


図 89 外気温度の頻度分布 (除湿)

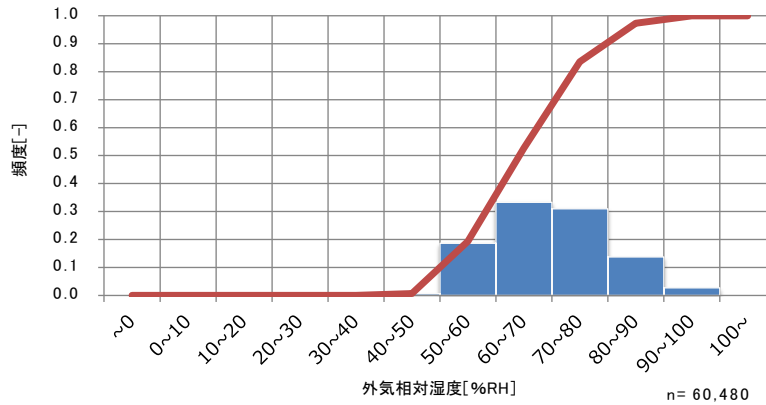


図 90 外気相対湿度の頻度分布 (除湿)

1.4 まとめ

1.4.1 温熱環境

温熱環境は、宮古島の外気温度は日変動が小さく 35℃を超えるような時間帯が無いいため室外と室内の温度差がほとんどない傾向があった。湿度に関してはエアコンによる除湿効果が見られた。また、エアコン使用時でも室内の相対湿度は 60 %RH 以上を推移しており機器稼働時においても多湿状態であることが分かった。

1.4.2 機器効率

機器効率は、外気温度の日変動が少なく最高気温が低いことから室外機の熱交換に有利な条件であるため COP が 10～30 の高い範囲で推移する傾向にあった。一方、処理熱量は定格 (=6.3 kW) 以下で推移しており低負荷での運用となったため、機器効率が低い範囲での使用が多い結果となった。

付録 E

家具の吸放湿特性実験

1 家具の吸放湿実験

1.1 目的

熱負荷計算における家具のモデル化の基礎データの取得を目的とする。

1.2 実験概要

実験室の湿度を周期的に低湿と高湿に制御し、家具の重量を測定した。

1.2.1 実験室の概要

実験室は建築研究所にある高精度温度湿度制御室を使用した。

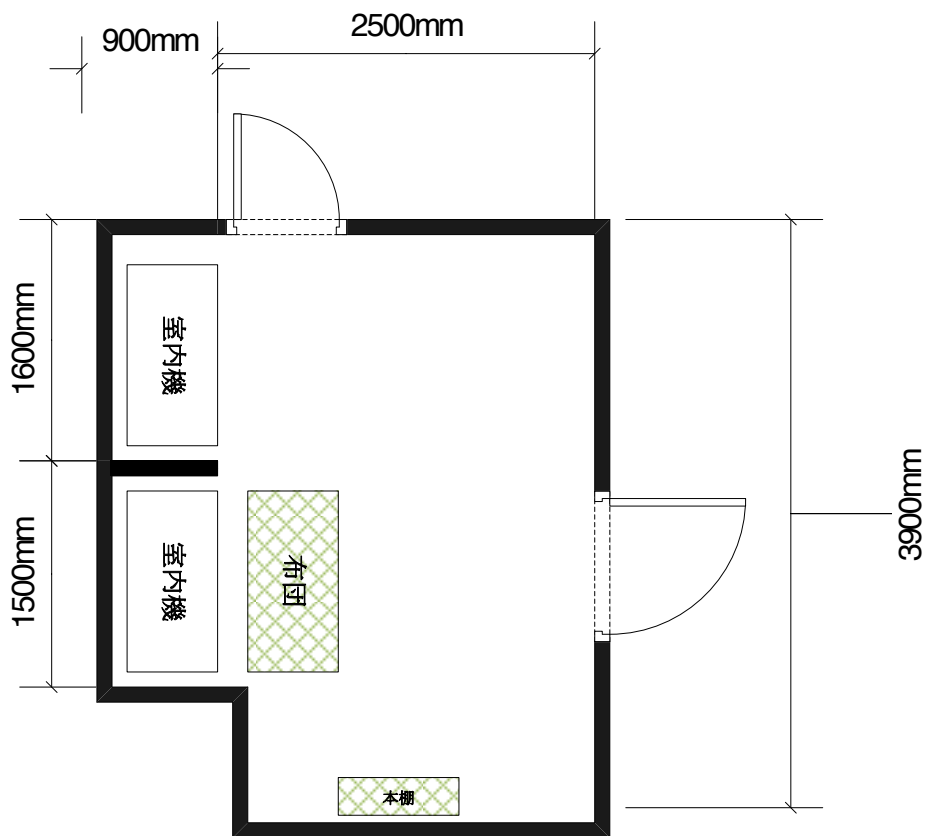


図 1 実験室平面図

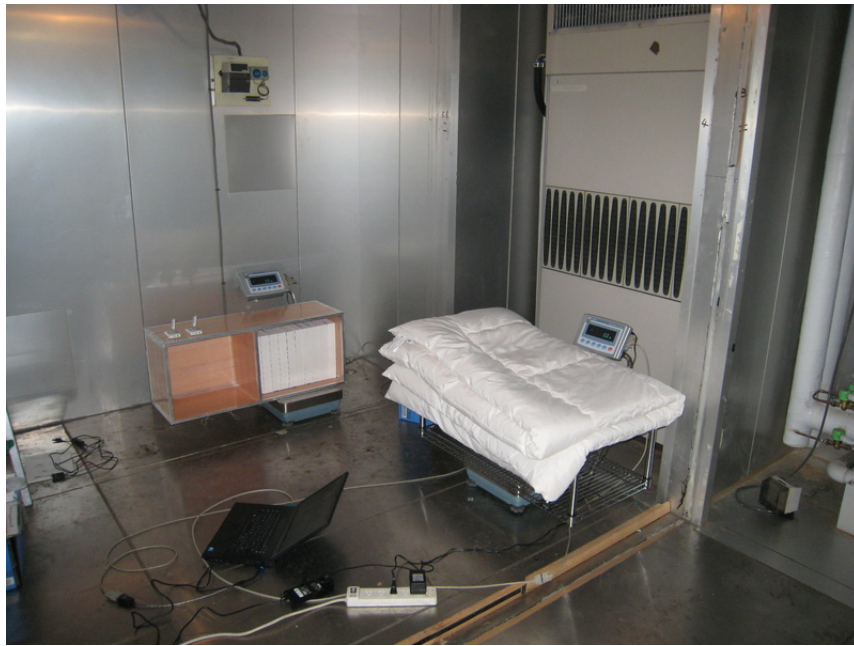


写真 1 実験室



写真 2 布団



写真 3 本棚

1.2.2 測定刻目

測定項目	対象	測定機器	補足
温度	家具内部、家具周辺、実験室内	熱電対	
湿度	家具内部、家具周辺、実験室内	抵抗変化式	
重量	本棚、布団	重量計	

1.2.3 実験条件

1.2.3.1 条件 1：相対湿度変化範囲が小さい

養生期間（低湿）	： 温度設定 30 °C 相対湿度設定 40 %RH	継続時間 141 時間
ステップ 1（高湿）	： 温度設定 20 °C 相対湿度設定 70 %RH	継続時間 24 時間
ステップ 2（低湿）	： 温度設定 30 °C 相対湿度設定 40 %RH	継続時間 24 時間
ステップ 3（低湿）	： 温度設定 20 °C 相対湿度設定 70 %RH	継続時間 24 時間
終了後（低湿）	： 温度設定 30 °C 相対湿度設定 40 %RH	

1.2.3.2 条件 2：相対湿度変化範囲が大きい

養生期間（低湿）	： 温度設定 40 °C 相対湿度設定 20 %RH	継続時間 227 時間
ステップ 1（高湿）	： 温度設定 15 °C 相対湿度設定 90 %RH	継続時間 48 時間
ステップ 2（低湿）	： 温度設定 40 °C 相対湿度設定 20 %RH	継続時間 48 時間
ステップ 3（低湿）	： 温度設定 15 °C 相対湿度設定 90 %RH	継続時間 48 時間
終了後（低湿）	： 温度設定 40 °C 相対湿度設定 20 %RH	

1.3 実験結果

1.3.1 条件 1：相対湿度変化の範囲が小さい

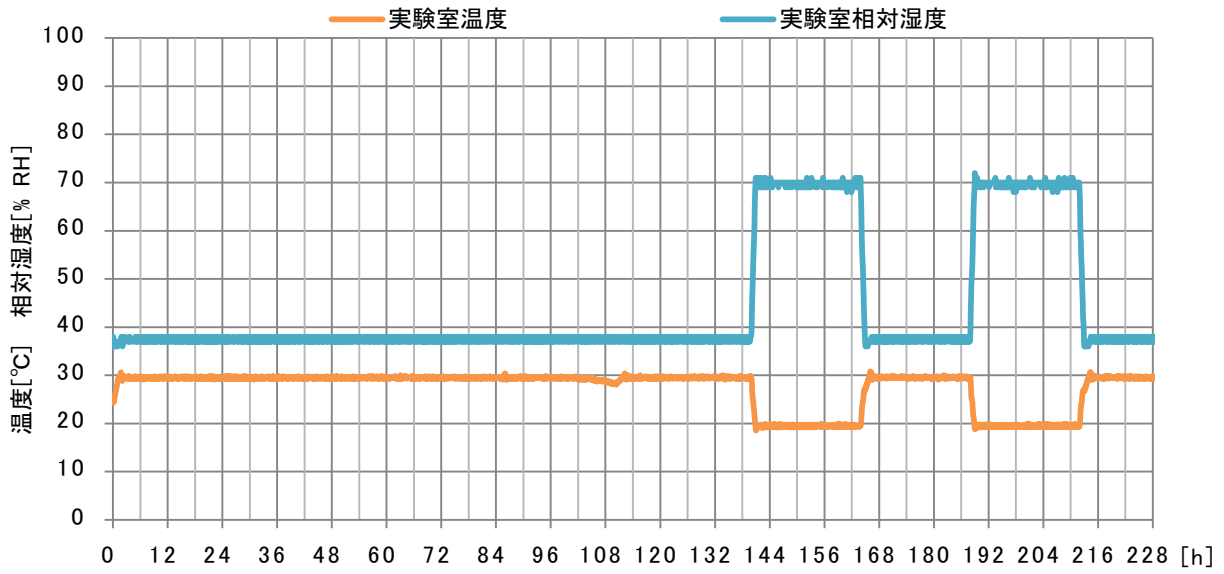


図 2 実験室の温熱環境の推移（相対湿度変化範囲小）

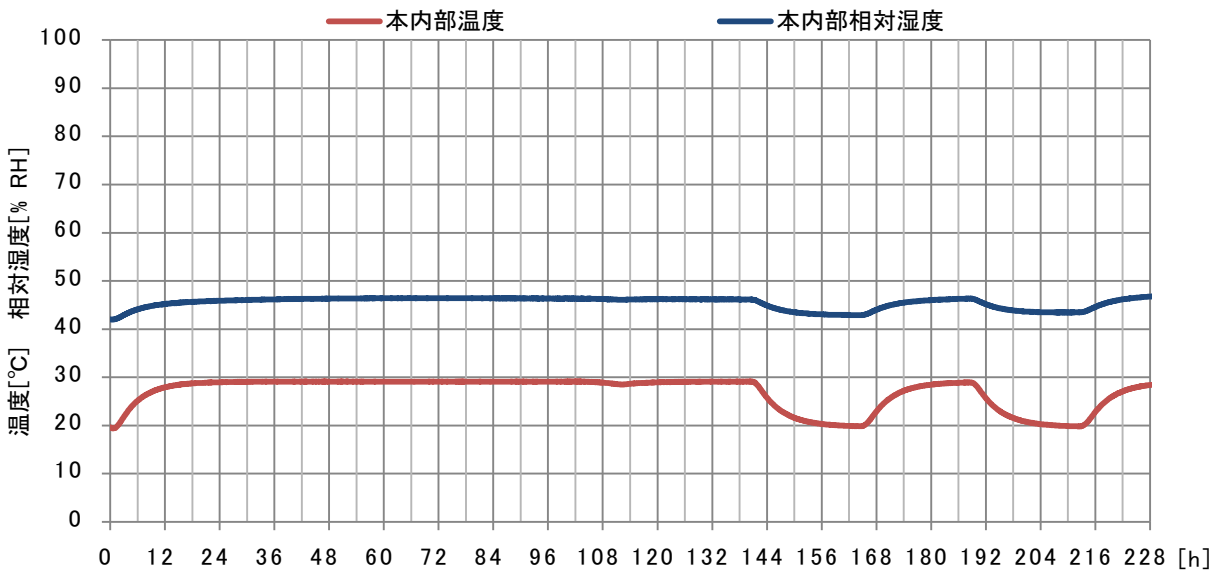


図 3 本内部の温湿度の推移（相対湿度変化範囲小）

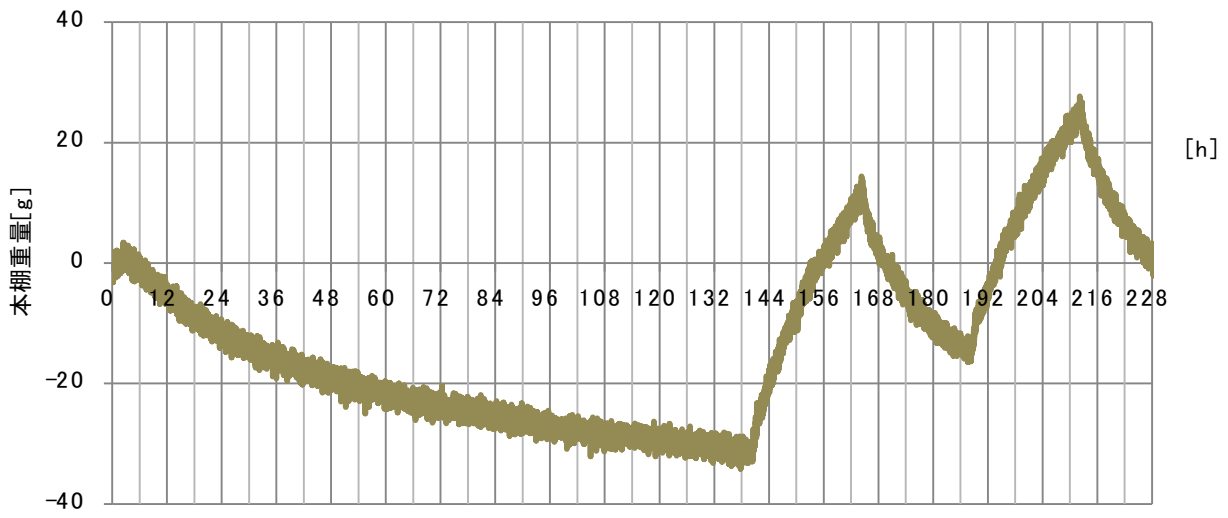


図 4 本棚の重量の推移 (相対湿度変化範囲小)

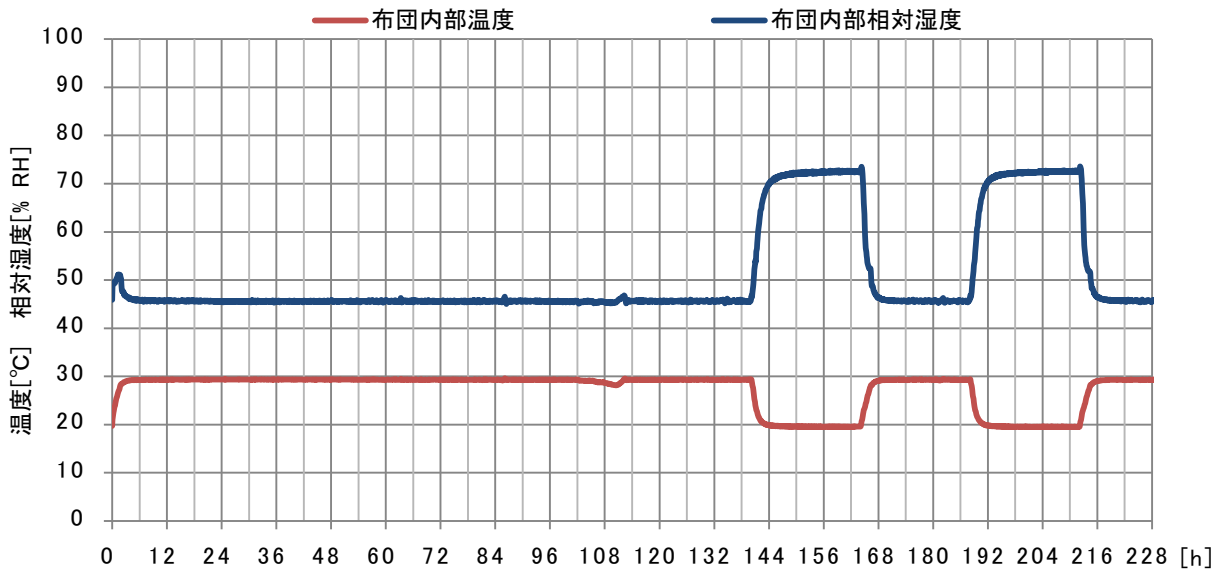


図 5 布団内部の温湿度の推移 (相対湿度変化範囲小)

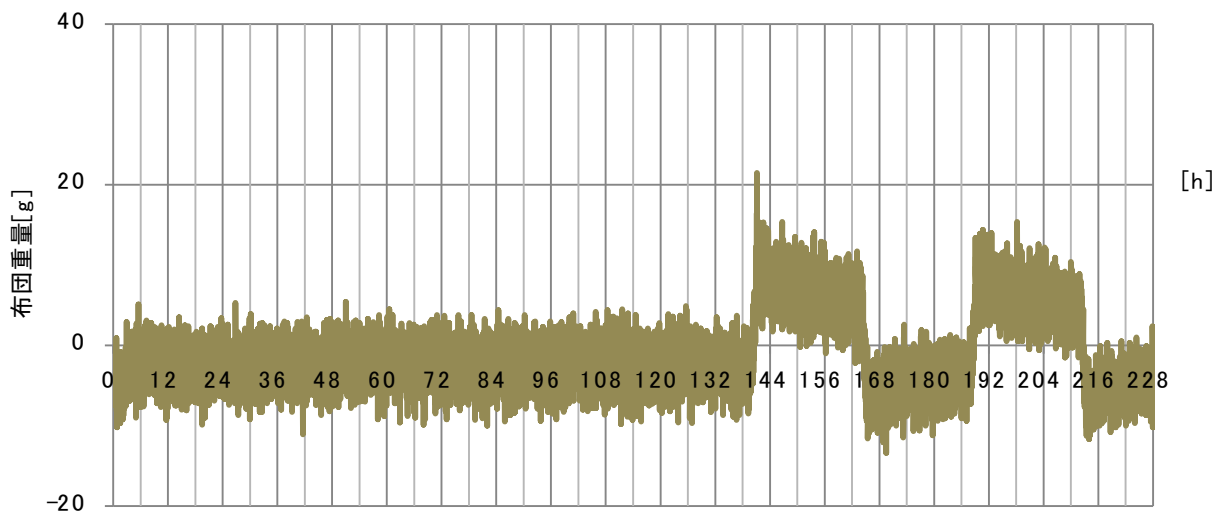


図 6 布団の重量の推移（相対湿度変化範囲小）

1.3.2 条件 2：相対湿度変化の範囲が大きい

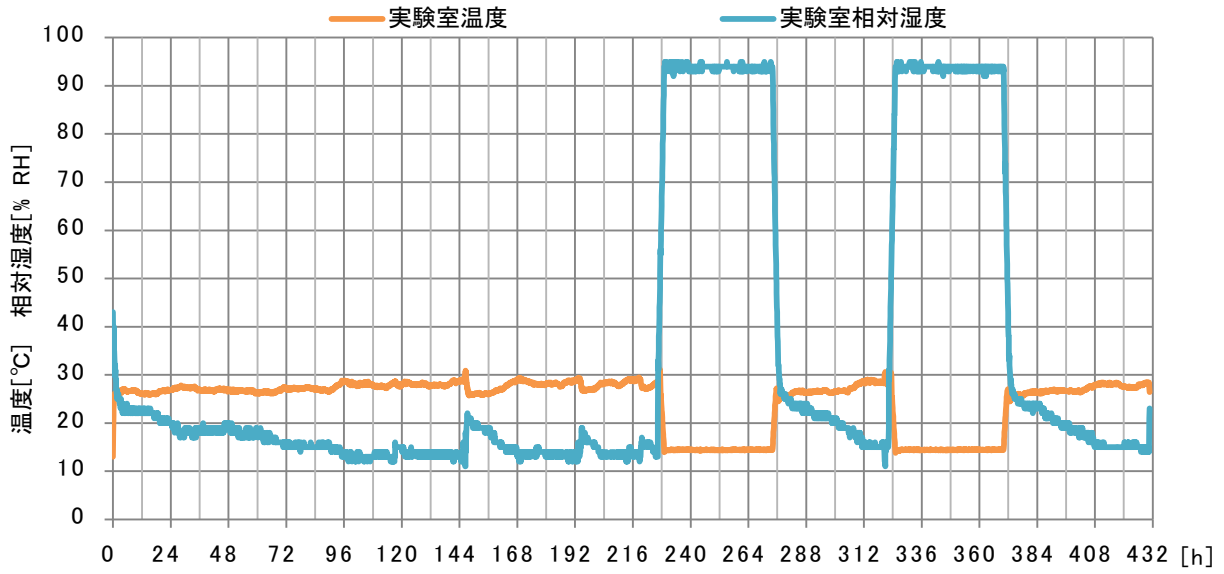


図 7 実験室の温熱環境の推移（相対湿度変化範囲大）

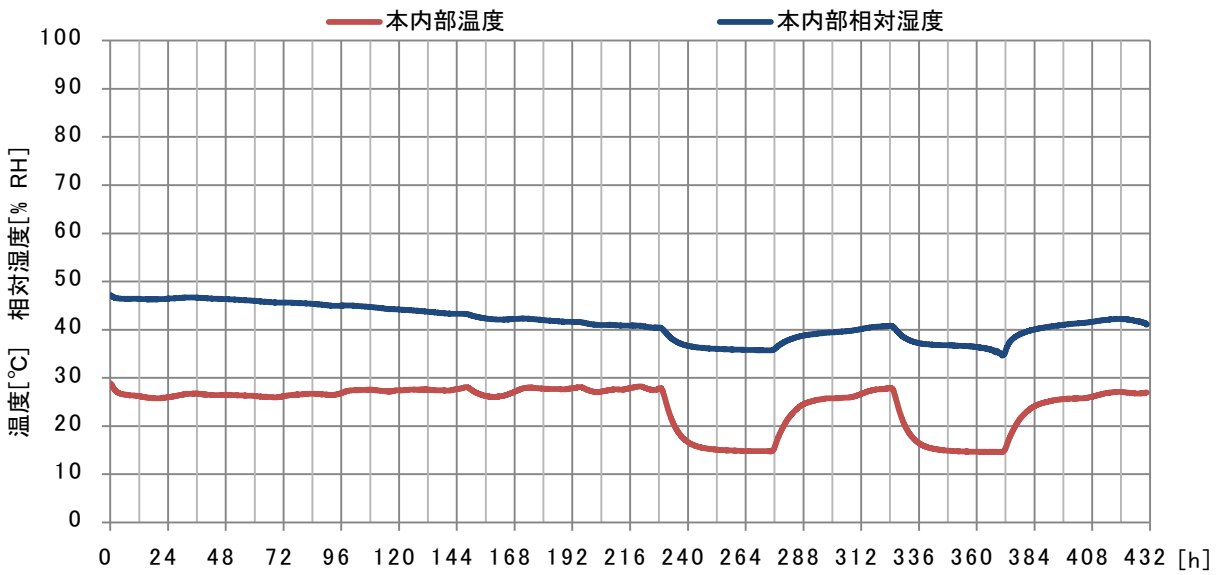


図 8 本内部の温湿度の推移（相対湿度変化範囲大）

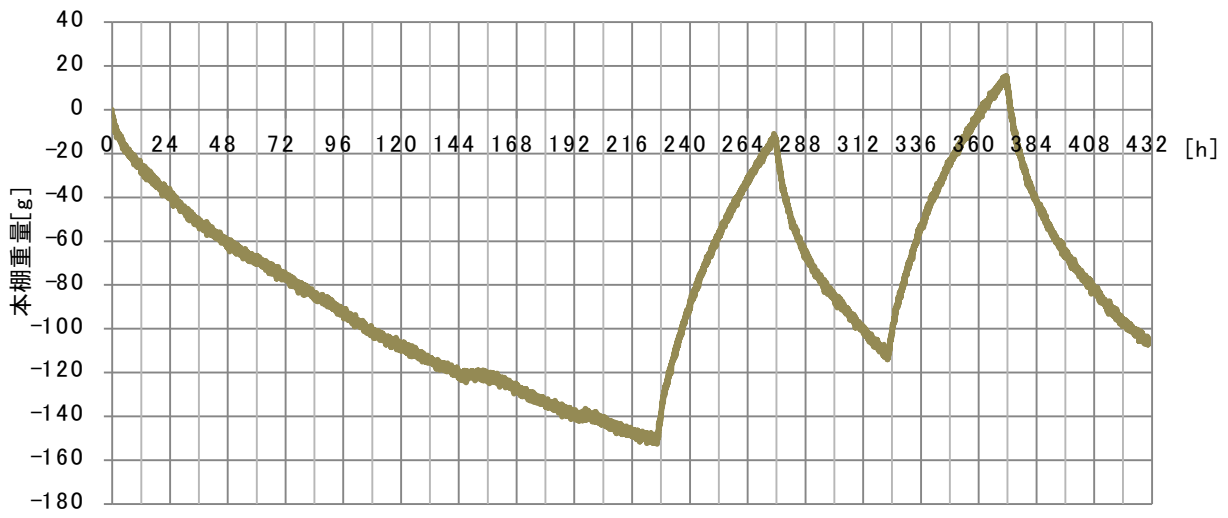


図 9 本棚の重量の推移（相对湿度変化範囲大）

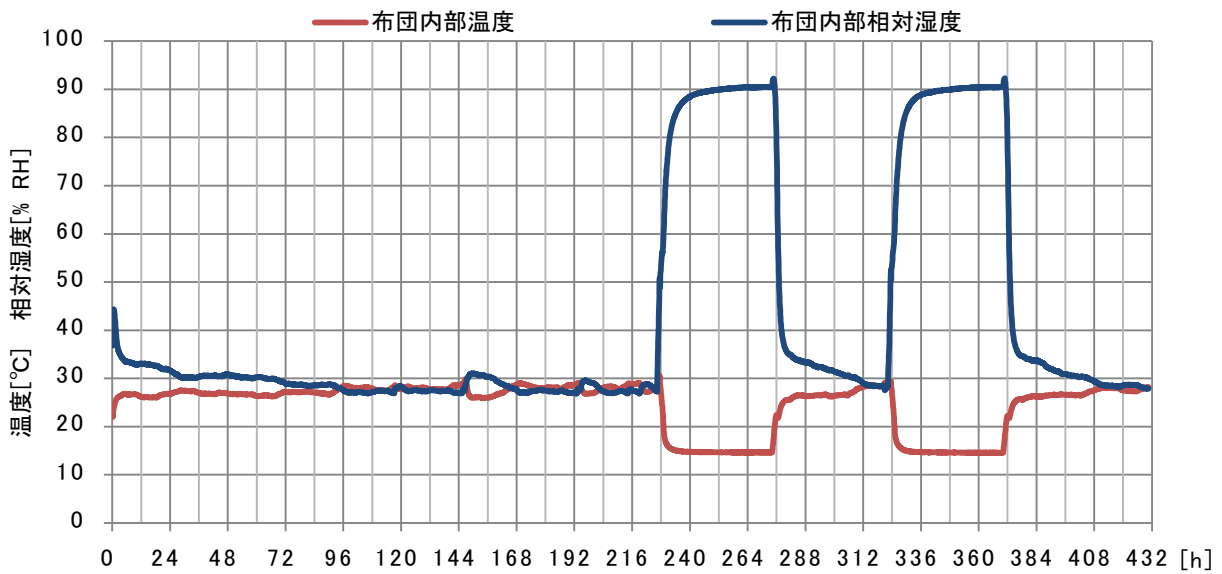


図 10 布団内部の温湿度の推移（相对湿度変化範囲大）

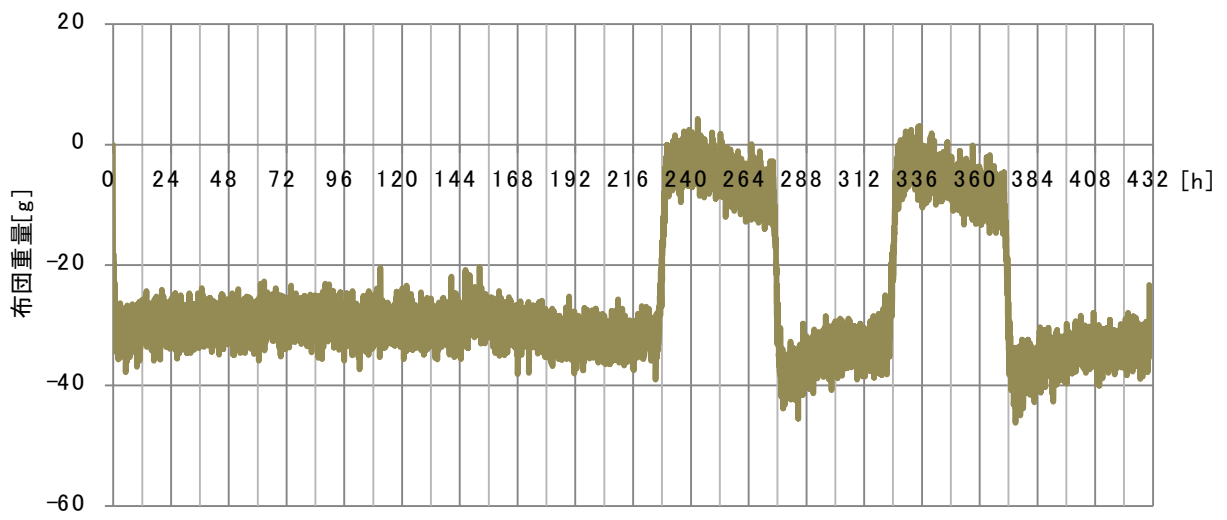


図 11 布団の重量の推移（相对湿度変化範囲大）

付録 F

プログラムソースコード

1 プログラムソースコード

1.1 Program.cs

```
using Excel;
using HLFX.Block;
using HLFX.Utility;
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Sample2
{
    class Program
    {
        static void Main(string[] args)
        {
            //ログファイル作成
            using (
                var log = new System.IO.StreamWriter(@"log.csv",
                    false, Encoding.GetEncoding(932))
            {
                var simulator = new Simulator()
                {
                    //計算時間、計算間隔
                    StartDay = new DateTime(2015, 4, 1, 0, 0, 0),
                    StopDay = new DateTime(2015, 10, 1, 0, 0, 0),
                    Dt = 60,

                    //ブロック設定
                    Blocks = BuildBlocks(),

                    //ログ出力
                    Logger = (context) =>
                    {
                        //出力間隔の設定
                        if ((context.Elapsed) % 3600 != 0) return;

                        StringBuilder sb = new StringBuilder();

                        sb.AppendFormat("{0},{1},", context.Current.Month
                            + "月" + context.Current.Day + "日",
                            context.Current.Hour + "時" +
                            context.Current.Minute + "分");

                        //状態出力
                        foreach (var obj in context.Blocks)
                        {
                            if (obj is WallBlock)
                            {
                                //var wall = obj as WallBlock;
                                //sb.AppendFormat("{0:F2}, {1:F2}, ",
                                    wall.TempA, wall.TempB);
                            }
                            else if (obj is WindowBlock)
                            {
                                //var wind = obj as WindowBlock;
                                //sb.AppendFormat("{0:F2}, {1:F2}, ",
                                    wind.TempA, wind.TempB);
                            }
                            else if (obj is FixedConditionRoomBlock)
                            {
                                var room = obj as FixedConditionRoomBlock;
                                sb.AppendFormat("{0:F2},", room.Temp);
                            }
                            else if (obj is RoomBlock)
                            {
                                var room = obj as RoomBlock;
                                sb.AppendFormat("{0:F2}, {1:F2}, {2:F2}, {3:F2}, {
                                    4:F2}, {5:F2}, {6:F2}, {7:F2}, ",
                                    room.Temp, room.MRT, room.Humidity * 1000.0,
                                    room.RHumidity * 100.0,
                                    room.SensibleHeatGenerationTotalConvection *
                                    3.6,
                                    room.VaporGenerationTotal * 1000.0 * 3600.0,
                                    (room.DirectSolarConvectionTotal +
                                    room.SkySolarConvectionTotal +
                                    room.ReflectSolarConvectionTotal) * 3.6,
                                    room.Qvent * 3.6);
                            }
                            else if (obj is OutsideBlock)
                            {
                                var _outside = obj as OutsideBlock;
                                sb.AppendFormat("{0:F2}, {1:F2}, ",
                                    _outside.Temp, _outside.Humidity * 1000.0);
                            }
                            else if (obj is AirConditionerBlock)
                            {
                                var rac = obj as AirConditionerBlock;
                                sb.AppendFormat("{0}, {1:F2}, {2:F2}, {3:F3}, {4:F
                                    6}, ", rac.Operation, rac.SensibleHeat,
                                    rac.LatentHeat * 2510.0 * 1000.0, rac.COP,
                                    rac.E_AC);
                            }
                            else if (obj is PersonBlock)
                            {
                                var person = obj as PersonBlock;
                                sb.AppendFormat("{0}, ", person.InRoom);
                            }
                        }

                        log.WriteLine(sb);

                        Console.Write("*");
                    }
                };

                //開始表示
                DateTime start = DateTime.Now;
                Console.WriteLine("開始: {0}", start);
                Console.WriteLine("計算間隔(壁): {0}秒",
                    simulator.Dt);
                Console.WriteLine("計算回数: {0}回",
                    simulator.GetTotalLoopCount());
                Console.WriteLine("=====");
                Console.WriteLine("=====");

                //ログヘッダー用意
                log.Write("日付,時刻,");
                foreach (var obj in simulator.Blocks)
                {
                    if (obj is FixedConditionRoomBlock)
                        log.Write("{0},", obj.Name);
                    else if (obj is RoomBlock)

```

```

log.Write("{0},,,,,,,", obj.Name);
//else if (obj is WallBlock || obj is WindowBlock)

//log.Write("{0},", obj.Name);
else if (obj is OutsideBlock)
log.Write("{0},", obj.Name);
else if (obj is AirConditionerBlock)
log.Write("{0},,,,,", obj.Name);
else if (obj is PersonBlock)
log.Write("{0},", obj.Name);
}
log.WriteLine("");
log.Write(", ");
foreach (var obj in simulator.Blocks)
{
//if (obj is WallBlock || obj is WindowBlock)
log.Write("A側温度,B側温度,");
if (obj is FixedConditionRoomBlock) log.Write("温度,");
else if (obj is RoomBlock) log.Write("温度,MRT,絶対湿度,相对湿度,内部発熱対流成分[kJ/h],潜熱発生[g/h],日射対流成分[kJ/h],換気流入熱[kJ/h],");
else if (obj is OutsideBlock) log.Write("温度,絶対湿度,");
else if (obj is AirConditionerBlock) log.Write("運転,顕熱[W],潜熱[W],COP[-],消費電力[kWh],");
else if (obj is PersonBlock) log.Write("在室,");
}
log.WriteLine("");

//計算
simulator.Run();

//終了表示
DateTime stop = DateTime.Now;
Console.WriteLine("=====");
Console.WriteLine("=====");
Console.WriteLine("終了: {0}", stop);
Console.WriteLine("経過時間: {0}ミリ秒", (stop - start).TotalMilliseconds);
Console.WriteLine("計算速度: {0}回/秒", (double)simulator.GetTotalLoopCount() / (stop - start).Seconds);
}

/// <summary>
/// ブロックの構築を行う
/// </summary>
/// <returns>構築済みのブロックのリスト</returns>
private static IList<IBlock> BuildBlocks()
{
double inittemp = 20.0;
double inithumidity = 7.354 / 1000.0; //20°C50%相当

var blocks = new List<IBlock>(); // IBlockを持つオブジェクトのリスト

using(FileStream stream = File.Open("input.xlsx", FileMode.Open, FileAccess.Read))
using (IExcelDataReader excelReader = ExcelReaderFactory.CreateOpenXmlReader(stream))
{
excelReader.IsFirstRowAsColumnNames = true;

```

```

DataSet data = excelReader.AsDataSet();

#region スケジュールリスト
var schedules = new Dictionary<string, ScheduleSeries>();
if (data.Tables["スケジュール"] != null)
{
foreach (DataRow row in data.Tables["スケジュール"].Rows)
{
string id = Convert.ToString(row[0]); //スケジュールID

if (!String.IsNullOrEmpty(id))
{
ScheduleSeries Schedule = new ScheduleSeries();

string sctype = Convert.ToString(row[1]);

for (int i = 0; i < 100; i++)
{
int time = Convert.ToInt32(row[2 * i + 2]);
if (time >= 2400) break;

double scval = Convert.ToDouble(row[2 * i + 3]);
if (sctype == "Light" || sctype == "Equip" || sctype == "Ventilation") //Light, Equip, Ventilationは[%]で入力
scval /= 100.0;

Schedule.UnitAdd(time, scval);
}

schedules.Add(id, Schedule);
}
}
}

#region 外部条件ブロック
var outside = new OutsideBlock() { FileName = @"C:\Program Files\AE-CAD\SimHeat\weather\6159999.sma" }; //最初に外部環境を生成
//キー(固定文字列)と値(outside)のセット
blocks.Add(outside);
#endregion

#region 空間ブロック
//var rooms = new Dictionary<string, IBlock>();
// キー(string:室名)と値(IBlock:RoomBlockのインスタンス)の組み合わせ
var rooms = new Dictionary<string, RoomBlock>();
foreach (DataRow row in data.Tables["居室"].Rows)
{
string No = Convert.ToString(row[0]);
if (No != "")
{
string Name = Convert.ToString(row[1]);
string 容積 = Convert.ToString(row[2]);
string 種別 = Convert.ToString(row[3]);
string 顕熱容量 = Convert.ToString(row[4]); //kJ/K
string 潜熱容量 = Convert.ToString(row[5]); //kg/(kg/kg')
string 換気1 = Convert.ToString(row[6]);

```

```

string 換気2 = Convert.ToString(row[7]);

if (顕熱容量 == "") 顕熱容量 = "0";
if (潜熱容量 == "") 潜熱容量 = "0";

if (!String.IsNullOrEmpty(No)
    && !String.IsNullOrEmpty(Name)
    && !String.IsNullOrEmpty(容積)
    && !String.IsNullOrEmpty(種別))
{
    rooms.Add(Name,
        new RoomBlock
        {
            Name = Name,
            Temp = inittemp,
            MRT = inittemp,
            Humidity = inithumidity,
            Cs = Convert.ToDouble(顕熱容量) * 1000.0,
            Cl = Convert.ToDouble(潜熱容量),
            VentNumClose = Convert.ToDouble(換気1),
            VentNumOpen = Convert.ToDouble(換気2),
            Volume = Convert.ToDouble(容積),
            TotalArea = 0.0,
            TotalFloorArea = 0.0,
            WindowOpen = 0,
            ACONOFF = 0,
            Outside = outside
        });
    //RoomBlockインスタンスの作成とroomsリストへの登録
}
}

blocks.AddRange(rooms.Select(x => x.Value)); //
ラムダ式 roomsに登録されたオブジェクトをすべて
blocksにも登録する

var fixedconditionrooms = new Dictionary<string,
FixedConditionRoomBlock>();
if (data.Tables["条件固定室"] != null)
{
    foreach (DataRow row in data.Tables["条件固定室"]
        .Rows)
    {
        string No = Convert.ToString(row[0]);
        if (No != "")
        {
            string Name = Convert.ToString(row[1]);

            IList<ISpaceBlock> referencespaces = new
                List<ISpaceBlock>();
            IList<double> ratios = new List<Double>();

            for (int i = 0; i < 100; i++)
            {
                string spacename = Convert.ToString(row[2 * i
                    + 2]);
                string ratio = Convert.ToString(row[2 * i +
                    3]);
                if (string.IsNullOrEmpty(ratio) || spacename
                    == "END")
                    break;
                else
                {
                    if (spacename == "Outside" || spacename ==
                        "")

```

```

{
    referencespaces.Add(outside);
    ratios.Add(Convert.ToDouble(ratio));
}

else if (rooms.ContainsKey(spacename))
{
    referencespaces.Add(rooms[spacename]);
    ratios.Add(Convert.ToDouble(ratio));
}
}

FixedConditionRoomBlock fixedconditionroom =
new FixedConditionRoomBlock(Name,
referencespaces, ratios);
fixedconditionrooms.Add(Name,
fixedconditionroom);
}
}
blocks.AddRange(fixedconditionrooms.Select(x =>
x.Value));

#endregion

#region 発熱体
var heatgenerators = new Dictionary<string,
HeatGenerator>();
if (data.Tables["発熱体"] != null)
{
    foreach (DataRow row in data.Tables["発熱体"]
        .Rows)
    {
        string No = Convert.ToString(row[0]);

        if (No != "")
        {
            string name = Convert.ToString(row[1]);
            string sensible = Convert.ToString(row[2]);
            string vapor = Convert.ToString(row[3]);
            // [kg/h]
            string convectionrate =
                Convert.ToString(row[4]);

            string[,] scheduleid = new string[3, 2];
            for (int term = 0; term < 3; term++)
            {
                for (int wday = 0; wday < 2; wday++)
                {
                    scheduleid[term, wday] =
                        Convert.ToString(row[2 * term + wday + 5]);
                    ///row[5]~[10]
                }
            }

            IList<RoomBlock> applyrooms = new
                List<RoomBlock>();

            ///適用室(複数)の取得
            for (int i = 0; i < 100; i++)
            {
                string roomname = Convert.ToString(row[11 +
                    i]);
                if (string.IsNullOrEmpty(roomname))
                    break;

```

```

else
{
    if (rooms.ContainsKey(roomname))
    {
        applyrooms.Add(rooms[roomname]);
    }
}
}

///スケジュール実体の取得
ScheduleSeries[,] scseries = new
ScheduleSeries[3, 2];
for (int term = 0; term < 3; term++)
{
    for (int wday = 0; wday < 2; wday++)
    {
        if (schedules.ContainsKey(scheduleid[term,
wday]))
            scseries[term, wday] =
schedules[scheduleid[term, wday]];
        else
            scseries[term, wday] = null;
    }
}

HeatGenerator heatgen = new HeatGenerator (name,
Convert.ToDouble(sensible),
Convert.ToDouble(vapor) / 3600.0,
Convert.ToDouble(convectionrate),
scseries, applyrooms);

heatgenerators.Add(No, heatgen);
}
}
}
#endregion

#region エアコン
var aircons = new Dictionary<string,
AirConditionerBlock>();
if (data.Tables["エアコン"] != null)
{
    foreach (DataRow row in data.Tables["エアコン
"].Rows)
    {
        string No = Convert.ToString(row[0]);
        if (No != "")
        {
            string name = Convert.ToString(row[1]);
            string roomname = Convert.ToString(row[2]);
            string heatingcapacity =
Convert.ToString(row[3]);
            string coolingcapacity =
Convert.ToString(row[4]);
            string airvolume = Convert.ToString(row[5]);
            string bypassfactor = Convert.ToString(row[6]);

            string[,] scheduleid = new string[3, 2];
            for (int term = 0; term < 3; term++)
            {
                for (int wday = 0; wday < 2; wday++)
                {
                    scheduleid[term, wday] =
Convert.ToString(row[2 * term + wday + 7]);
                    ///row[7]~[12]

```

```

}
}

string worktype = Convert.ToString(row[13]);

RoomBlock room = new RoomBlock();
if (rooms.ContainsKey(roomname))
    room = rooms[roomname];

///スケジュール実体の取得
ScheduleSeries[,] scseries = new
ScheduleSeries[3, 2];
for (int term = 0; term < 3; term++)
{
    for (int wday = 0; wday < 2; wday++)
    {
        if (schedules.ContainsKey(scheduleid[term,
wday]))
            scseries[term, wday] =
schedules[scheduleid[term, wday]];
        else
            scseries[term, wday] = null;
    }
}

AirConditionerBlock ac = new
AirConditionerBlock(
name, room, outside,
Convert.ToDouble(heatingcapacity),
Convert.ToDouble(coolingcapacity),
Convert.ToDouble(airvolume) / 3600.0,
Convert.ToDouble(bypassfactor), scseries);

blocks.Add(ac);
}
}
}
#endregion

#region 在室者
var persons = new Dictionary<string,
PersonBlock>();
if (data.Tables["在室者"] != null)
{
    foreach (DataRow row in data.Tables["在室者
"].Rows)
    {
        string No = Convert.ToString(row[0]);
        //
        if (No != "")
        {
            string total = Convert.ToString(row[1]);
            //一人当たり最大全熱量[W/人]
            string stdsensible = Convert.ToString(row[2]);
            //一人当たり顕熱発熱量[W/人], 24℃の場合
            string slope = Convert.ToString(row[3]);
            //
            string convectionrate =
Convert.ToString(row[4]); //対流成分比率

            string[,] scheduleid = new string[3, 2];
            ///スケジュールID
            for (int term = 0; term < 3; term++)
            {

```



```

    for (int wday = 0; wday < 2; wday++)
    {
        scheduleid[term, wday] =
            Convert.ToString(row[2 * term + wday + 5]);
        //row[5]~[10]
    }
}

string roomname = Convert.ToString(row[11]);
RoomBlock room = new RoomBlock();
if (rooms.ContainsKey(roomname))
    room = rooms[roomname];

//スケジュール実体の取得
ScheduleSeries[,] scseries = new
ScheduleSeries[3, 2];
for (int term = 0; term < 3; term++)
{
    for (int wday = 0; wday < 2; wday++)
    {
        if (schedules.ContainsKey(scheduleid[term,
wday]))
            scseries[term, wday] =
                schedules[scheduleid[term, wday]];
        else
            scseries[term, wday] = null;
    }
}

PersonBlock person = new PersonBlock(
    Convert.ToDouble(total),
    Convert.ToDouble(stdsensible),
    Convert.ToDouble(slope),
    Convert.ToDouble(convectionrate), scseries,
    room);

blocks.Add(person);
}
}
}

#endregion

#region 材料テーブル
var materials = new Dictionary<string, Material>();
//キー(材料No)と値(Materialのインスタンス)の
//組み合わせ
foreach (DataRow row in data.Tables["材料"].Rows)
{
    string No = Convert.ToString(row[0]);
    // 材料No.
    string Name = Convert.ToString(row[1]);
    // 材料名
    string 熱伝導率 = Convert.ToString(row[2]);
    // [W/mK]
    string 空気層熱抵抗 = Convert.ToString(row[3]);
    // [m2K/W]
    string 容積比率 = Convert.ToString(row[4]);
    // [kJ/m3K]
    string 密度 = Convert.ToString(row[5]);
    // [kg/m3]
    if (!String.IsNullOrEmpty(No)
        && !String.IsNullOrEmpty(Name)
        && !String.IsNullOrEmpty(熱伝導率)
        && !String.IsNullOrEmpty(空気層熱抵抗)
        && !String.IsNullOrEmpty(容積比率)
        && !String.IsNullOrEmpty(密度))

```

```

{
    materials.Add(No,
        new Material
        {
            Name = Name,
            熱伝導率 = Convert.ToDouble(熱伝導率),
            空気層熱抵抗 = Convert.ToDouble(空気層熱抵抗),
            容積比熱 = Convert.ToDouble(容積比率) *
                1000.0,
            密度 = Convert.ToDouble(密度)
        });
    //材
    //料インスタンスの作成とmaterialsへの登録
}
}

#endregion

#region 壁仕様
var wallSpecs = new Dictionary<string, WallSpec>();
//キー(層構成名)と値(WallSpec)の組み合わせ
string No_old = "";
string No_new = "";
foreach (DataRow row in data.Tables["壁仕様"].Rows)
{
    No_old = No_new;
    No_new = Convert.ToString(row[0]);
    // 壁仕様No.
    string secname = Convert.ToString(row[1]);
    // 断面名・No.
    string name = Convert.ToString(row[2]);
    // 壁仕様名
    string secratio = Convert.ToString(row[3]);
    // 断面比率
    bool newspec = true;
    // 新規仕様かどうか

    if (No_old == No_new) newspec = false;

    if (!String.IsNullOrEmpty(name))
    {
        int count = Convert.ToInt32(row[4]); //材料
        //数

        // 断面の登録
        var wallsection = new WallSection(secname,
            Convert.ToDouble(secratio), count);
        for (int i = 0; i < count; i++)
        {
            wallsection.materiallist.Add(new
                WallMaterialList
                {
                    Material = materials[Convert.ToString(row[5 +
                        i * 2])],
                    Thickness = Convert.ToDouble(row[6 + i * 2]) /
                        1000.0
                });
        }

        // 層構成の新規生成
        if (newspec == true)
        {
            wallSpecs.Add(name,
                new WallSpec
                {
                    Name = name,
                    SectionNum = 0
                });
        }
    }
}
}

```

```

}

WallSpec wallspec = wallSpecs[name];
wallspec.SectionNum++;
wallspec.wallsection.Add(wallsection);
}
}
#endregion

#region 窓仕様
var windowProperties = new Dictionary<string,
WindowProperty>();
foreach (DataRow row in data.Tables["窓仕様"].Rows)
{
string No = Convert.ToString(row[0]);
string 熱貫流率 = Convert.ToString(row[1]);
string  $\eta R$  = Convert.ToString(row[2]);
string  $\eta C$  = Convert.ToString(row[3]);
string 厚さ = Convert.ToString(row[4]);

if (!String.IsNullOrEmpty(No)
&& !String.IsNullOrEmpty(熱貫流率)
&& !String.IsNullOrEmpty( $\eta R$ )
&& !String.IsNullOrEmpty( $\eta C$ )
&& !String.IsNullOrEmpty(厚さ))
{
windowProperties.Add(No,
new WindowProperty
{
UValue = Convert.ToDouble(熱貫流率),
EtaR = Convert.ToDouble( $\eta R$ ),
EtaC = Convert.ToDouble( $\eta C$ ),
Thickness = Convert.ToDouble(厚さ) / 1000.0
});
}
}

#endregion

#region 壁ブロック
//複数断面の壁は、1断面ごとにインスタンスを作成する
var walls = new Dictionary<string, IBlock>(); //
キー(string:No.)と値(IBlock:WallBlockのインスタンス)
の組み合わせ
foreach (DataRow row in data.Tables["壁"].Rows)
{
string No = Convert.ToString(row[0]);
if (No != "")
{
string 居室A = Convert.ToString(row[1]);
string 居室B = Convert.ToString(row[2]); //
空白の場合は外部環境
double 方位角 = Convert.ToDouble(row[3]);
double 傾斜角 = Convert.ToDouble(row[4]);
string 面積 = Convert.ToString(row[5]);
string 床フラグ = Convert.ToString(row[6]);
string 壁仕様名 = Convert.ToString(row[7]);
string 壁仕様向き = Convert.ToString(row[8]);
string A側 $\alpha c$  = Convert.ToString(row[9]);
string A側 $\alpha r$  = Convert.ToString(row[10]);
string A側日射吸収率 = Convert.ToString(row[11]);
string A側長波放射率 = Convert.ToString(row[12]);
string B側 $\alpha c$  = Convert.ToString(row[13]);
string B側 $\alpha r$  = Convert.ToString(row[14]);

```

```

string B側日射吸収率 = Convert.ToString(row[15]);
string B側長波放射率 = Convert.ToString(row[16]);

if (!String.IsNullOrEmpty(No)
&& !String.IsNullOrEmpty(居室A)
&& !String.IsNullOrEmpty(面積)
&& !String.IsNullOrEmpty(壁仕様名)
&& !String.IsNullOrEmpty(壁仕様向き))
{
var roomA = rooms[居室A] as RoomBlock;

ISpaceBlock roomB = null;
if (居室B == "")
{
居室B = "外気";
roomB = outside;
}
else if (rooms.ContainsKey(居室B))
{
roomB = rooms[居室B] as RoomBlock;
}

else if (fixedconditionrooms.ContainsKey(居室B))
{
roomB = fixedconditionrooms[居室B] as
FixedConditionRoomBlock;
}

//断面ごとにブロック作成
WallSpec wallspec = new WallSpec();
wallspec = wallSpecs[壁仕様名];
int secnum = wallspec.SectionNum;
double totalratio = 0.0;
for (int i = 0; i < secnum; i++)
totalratio +=
wallspec.wallsection[i].SectionAreaRatio;

for (int i = 0; i < secnum; i++)
{
WallSection wallsection = new WallSection();
wallsection = wallspec.wallsection[i];
double ratio = wallsection.SectionAreaRatio;
string secname = wallsection.SecName;

//壁
var wall = new WallBlock(
name: String.Format("{0}-{1}-{2}-{3}", No,
居室A, 居室B, 壁仕様名),
spaceA: roomA,
spaceB: roomB,
area: Convert.ToDouble(面積) * ratio /
totalratio,
floorofspaceA: Convert.ToInt32(床フラグ),
materialList:
wallsection.materialList.Select(x =>
x.Material).ToArray(),
thicknessList:
wallsection.materialList.Select(x =>
x.Thickness).ToArray(),
matdirection: Convert.ToInt32(壁仕様向き),
alphacA: Convert.ToDouble(A側 $\alpha c$ ),
alpharA: Convert.ToDouble(A側 $\alpha r$ ),
solarabsorptionrateA: Convert.ToDouble(A側日
射吸収率),
longwaveemissionrateA: Convert.ToDouble(A側

```

```

    長波放射率),
    alphacB: Convert.ToDouble(B側 $\alpha c$ ),
    alphanB: Convert.ToDouble(B側 $\alpha r$ ),
    solarabsorptionrateB: Convert.ToDouble(B側日
    射吸収率),
    longwaveemissionrateB: Convert.ToDouble(B側
    長波放射率),
    faceh: Convert.ToDouble(傾斜角),
    facea: Convert.ToDouble(方位角)
);
//WallBlockインスタンスの生成
walls.Add(No + secname, wall);
//wallsリストへの登録

//A側
blocks.Add(wall.FaceA);

if (roomA is FixedConditionRoomBlock)
{
    blocks.Add(new
    TotalHeatTransferBlock(wall.FaceA));
}
else
{
    blocks.Add(new
    ConvectionHeatTransferBlock(wall.FaceA));
    blocks.Add(new
    ConvectionMoistureTransferBlock(wall.FaceA))
;
}

blocks.Add(wall.FaceB);
if (roomB is OutsideBlock) //B側が外
部
{
    blocks.Add(new
    TotalHeatTransferBlock(wall.FaceB));
}
else if (roomB is FixedConditionRoomBlock)
{
    blocks.Add(new
    TotalHeatTransferBlock(wall.FaceA));
}
else
{
    blocks.Add(new
    ConvectionHeatTransferBlock(wall.FaceB));
    blocks.Add(new
    ConvectionMoistureTransferBlock(wall.FaceB))
;
}
}
}
}
}
blocks.AddRange(walls.Select(x => x.Value));
//wallsに登録されたオブジェクトをすべてblocksにも登
録する
#endregion

#region 窓ブロック
var windows = new Dictionary<string, IBlock>();
// キー(string:No.)と値(IBlock:WindowBlockのインスタ
ンス)の組み合わせ
if (data.Tables["窓"] != null)
{
    foreach (DataRow row in data.Tables["窓"].Rows)

```

```

{
    string No = Convert.ToString(row[0]);
    if (No != "")
    {
        string 居室A = Convert.ToString(row[1]);
        string 居室B = Convert.ToString(row[2]);
        string 方位角 = Convert.ToString(row[3]);
        string 傾斜角 = Convert.ToString(row[4]);
        string 縦寸法 = Convert.ToString(row[5]);
        ///[m]
        string 横寸法 = Convert.ToString(row[6]);
        ///[m]
        string 窓仕様No = Convert.ToString(row[7]);
        ///Noで指定

        if (!String.IsNullOrEmpty(No)
            && !String.IsNullOrEmpty(居室A)
            && !String.IsNullOrEmpty(縦寸法)
            && !String.IsNullOrEmpty(横寸法)
            && !String.IsNullOrEmpty(窓仕様No))
        {
            //居室A,B
            var roomA = rooms[居室A] as RoomBlock;
            ISpaceBlock roomB;
            if (居室B == "")
            {
                居室B = "外気";
                roomB = outside;
            }
            else
            {
                roomB = rooms[居室B] as RoomBlock;
            }

            //窓
            var window = new WindowBlock(
                name: String.Format("Win{0}-{1}-{2}", No, 居
                室A, 居室B),
                spaceA: roomA,
                spaceB: roomB,
                width: Convert.ToDouble(縦寸法),
                height: Convert.ToDouble(横寸法),
                uvalue: windowProperties[窓仕様No].UValue,
                etar: windowProperties[窓仕様No].EtaR,
                etac: windowProperties[窓仕様No].EtaC,
                thickness: windowProperties[窓仕様
                No].Thickness,
                faceh: Convert.ToDouble(傾斜角),
                facea: Convert.ToDouble(方位角)
            );
            windows.Add(No, window);

            //A側は必ず居室
            blocks.Add(window.FaceA);
            blocks.Add(new
            ConvectionHeatTransferBlock(window.FaceA));
            blocks.Add(new
            ConvectionMoistureTransferBlock(window.FaceA))
;

            blocks.Add(window.FaceB);
            if (roomB is OutsideBlock) //B側が外
            部
            {
                blocks.Add(new
                TotalHeatTransferBlock(window.FaceB));
            }
        }
    }
}

```

```

    }
    else
    {
        blocks.Add(new
            ConvectionHeatTransferBlock(window.FaceB));
        blocks.Add(new
            ConvectionMoistureTransferBlock(window.FaceB
        ));
    }
}
}
}
blocks.AddRange(windows.Select(x => x.Value));
}
#endregion

#region ドアブロック
var doors = new Dictionary<string, IBlock>();
if (data.Tables["ドア"] != null)
{
    foreach (DataRow row in data.Tables["ドア"].Rows)
    {
        string No = Convert.ToString(row[0]);
        if (No != "")
        {
            string 居室A = Convert.ToString(row[1]);
            string 居室B = Convert.ToString(row[2]);
            string 方位角 = Convert.ToString(row[3]);
            string 傾斜角 = Convert.ToString(row[4]);
            string 縦寸法 = Convert.ToString(row[5]);
            ///[m]
            string 横寸法 = Convert.ToString(row[6]);
            ///[m]
            string 熱貫流率 = Convert.ToString(row[7]);
            ///
            string 材料No = Convert.ToString(row[8]);
            string 材料厚さ = Convert.ToString(row[9]);
            ///[mm]

            if (!String.IsNullOrEmpty(No)
                && !String.IsNullOrEmpty(居室A)
                && !String.IsNullOrEmpty(縦寸法)
                && !String.IsNullOrEmpty(横寸法))
            {
                //居室A,B
                var roomA = rooms[居室A] as RoomBlock;
                ISpaceBlock roomB;
                if (居室B == "")
                {
                    居室B = "外気";
                    roomB = outside;
                }
                else
                {
                    roomB = rooms[居室B] as RoomBlock;
                }

                Material material = materials[材料No];

                var door = new DoorBlock(
                    name: String.Format("Dor {0}-{1}-{2}", No, 居
                        室A, 居室B),
                    spaceA: roomA,
                    spaceB: roomB,
                    width: Convert.ToDouble(縦寸法),
                    height: Convert.ToDouble(横寸法),
                    uvalue: Convert.ToDouble(熱貫流率),

```

```

                    face: Convert.ToDouble(傾斜角),
                    facea: Convert.ToDouble(方位角),
                    material: material,
                    thickness: Convert.ToDouble(材料厚さ) /
                        1000.0
                );
                doors.Add(No, door);

                //A側は必ず居室
                blocks.Add(door.FaceA);
                blocks.Add(new
                    ConvectionHeatTransferBlock(door.FaceA));
                blocks.Add(new
                    ConvectionMoistureTransferBlock(door.FaceA));

                blocks.Add(door.FaceB);

                if (roomB is OutsideBlock) //B側が外部
                {
                    blocks.Add(new
                        TotalHeatTransferBlock(door.FaceB));
                }
                else
                {
                    blocks.Add(new
                        ConvectionHeatTransferBlock(door.FaceB));
                    blocks.Add(new
                        ConvectionMoistureTransferBlock(door.FaceB)
                    );
                }
            }
        }
        blocks.AddRange(doors.Select(x => x.Value));
    }
}
#endregion

#region 換気ブロック
var ventilations = new Dictionary<String,
    IBlock>();
if (data.Tables["換気"] != null)
{
    foreach (DataRow row in data.Tables["換気"].Rows)
    {
        string No = Convert.ToString(row[0]);
        if (No != "")
        {
            string 居室A = Convert.ToString(row[1]);
            //空白の場合は外部環境
            string 居室B = Convert.ToString(row[2]);
            //空白の場合は外部環境
            string 風量 = Convert.ToString(row[3]);

            string[,] scheduleid = new string[3, 2];
            for (int term = 0; term < 3; term++)
            {
                for (int wday = 0; wday < 2; wday++)
                {
                    scheduleid[term, wday] =
                        Convert.ToString(row[2 * term + wday + 4]);
                    ///row[4]~[9]
                }
            }

```

```

ISpaceBlock roomA = null;
ISpaceBlock roomB = null;
if (居室A == "" || 居室A == "Outside")
{
    居室A = "外気";
    roomA = outside;
}
else if (fixedconditionrooms.ContainsKey(居室A))
{
    roomA = fixedconditionrooms[居室A] as
    FixedConditionRoomBlock;
}
else
{
    roomA = rooms[居室A] as RoomBlock;
}

if (居室B == "" || 居室B == "Outside")
{
    居室B = "外気";
    roomB = outside;
}
else if (fixedconditionrooms.ContainsKey(居室B))
{
    roomA = fixedconditionrooms[居室B] as
    FixedConditionRoomBlock;
}
else
{
    roomB = rooms[居室B] as RoomBlock;
}

///スケジュール実体の取得
ScheduleSeries[,] scseries = new
ScheduleSeries[3, 2];
for (int term = 0; term < 3; term++)
{
    for (int wday = 0; wday < 2; wday++)
    {
        if (schedules.ContainsKey(scheduleid[term,
        wday]))
            scseries[term, wday] =
            schedules[scheduleid[term, wday]];
        else
            scseries[term, wday] = null;
    }
}

var ventilation = new
VentilationHeatTransferBlock(
    upperspace: roomA,
    lowerspace: roomB,
    airvolume: Convert.ToDouble(風量) / 3600.0,
    schedule: scseries
);
ventilations.Add(No, ventilation);
}
}
blocks.AddRange(ventilations.Select(x =>
x.Value));
}
}
#endregion

```

```

excelReader.Close();

return blocks;
}
}
}
}

```

1.2 Simulator.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace HLFX
{
    using Block;

    public class Simulator
    {
        /// <summary>
        /// 計算時間間隔[s]
        /// 原則60s
        /// 壁,窓などはさらに細かい時間間隔で行う
        /// </summary>
        public long Dt { get; set; }

        /// <summary>
        /// 開始日時
        /// </summary>
        public DateTime StartDay { get; set; }

        /// <summary>
        /// 終了日時
        /// </summary>
        public DateTime StopDay { get; set; }

        public IList<IBlock> Blocks { get; set; }

        /// <summary>
        /// ロガー
        /// </summary>
        public Action<SimulationContext> Logger { get; set; }

        public Simulator()
        {
            Dt = 60; //初期値60s

            StartDay = new DateTime(2015, 7, 5, 0, 0, 0);
            StopDay = new DateTime(2015, 7, 31, 23, 50, 0);
        }

        public void Run()
        {
            long elapsed = 0L;
            long totalClock = GetTotalLoopCount();

            var context = new SimulationContext()
            {

```

```

Dt = Dt,
Elapsed = 0L,
Current = StartDay,
Before = StartDay,
Blocks = Blocks,
};

DateTime heatingtermstartday = new DateTime(2014, 11,
30);
DateTime heatingtermendday = new DateTime(2015, 1,
2);
DateTime coolingtermstartday = new DateTime(2015, 6,
1);
DateTime coolingtermendday = new DateTime(2015, 9,
30);
context.SetTerm(heatingtermstartday,
heatingtermendday, coolingtermstartday,
coolingtermendday);

InitBlocks(context);

for (long i = 0; i < totalClock; i++, elapsed += Dt)
{
    context.Before = context.Current;
    context.Current = StartDay.AddSeconds(elapsed);
    context.Elapsed = elapsed;

    ///
    /// 気象データ更新
    /// 外部条件は必ず最初のブロックとしておく
    ///
    Blocks[0].IterationInit(context);

    ///
    /// 表面への日射セット
    ///
    foreach (var kernel in Blocks)
    {
        if (kernel is WallFaceConnector)
            kernel.IterationInit(context);
    }

    ///
    /// 窓の透過日射計算
    ///
    foreach (var kernel in Blocks)
    {
        if (kernel is WindowBlock)
            kernel.IterationInit(context);
    }

    ///
    /// 室の内部発熱・除去熱量計算(エアコン判定含む)
    ///
    foreach (var kernel in Blocks)
    {
        if (kernel is RoomBlock)
            kernel.IterationInit(context);
    }

    ///
    /// 条件固定室の温度計算
    ///
    foreach (var kernel in Blocks)
    {
        if (kernel is FixedConditionRoomBlock)
            kernel.IterationInit(context);

```

```

}

///
/// エアコン処理熱量熱量計算
///
foreach (var kernel in Blocks)
{
    if (kernel is AirConditionerBlock)
        kernel.Run(context);
}

///
/// 対流・総合熱伝達
///
foreach (var kernel in Blocks)
{
    if (kernel is ConvectionHeatTransferBlock ||
kernel is TotalHeatTransferBlock)
        ///
        /// WallFaceのQcに値をセットする
        ///
        kernel.Run(context);
}

///
/// 換気
///
foreach (var kernel in Blocks)
{
    if (kernel is VentilationHeatTransferBlock)
        kernel.Run(context);
}

///
/// 窓の伝熱計算
/// 透過日射は計算済み
///
foreach (var kernel in Blocks)
{
    if (kernel is WindowBlock)
        kernel.Run(context);
}

///
/// ドアの伝熱計算
///
foreach (var kernel in Blocks)
{
    if (kernel is DoorBlock)
        kernel.Run(context);
}

///
/// 壁の計算
///
foreach (var kernel in Blocks)
{
    if (kernel is WallBlock)
        kernel.Run(context);
}

///
/// 居室の計算
///
foreach (var kernel in Blocks)

```

```

    {
        if (kernel is RoomBlock)
            kernel.Run(context);
    }

    if (Logger != null)
    {
        Logger(context);
    }

    foreach (var obj in Blocks)
    {
        obj.Commit(context);
    }
}

private void InitBlocks(SimulationContext context)
{
    foreach (var block in context.Blocks)
    {
        block.Init(context);
    }
    int n = 0;
}

public long GetTotalLoopCount()
{
    return (long)Math.Ceiling((StopDay -
        StartDay).TotalSeconds / Dt);
}
}
}

```

1.3 SimulationContext.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using HLFX.Block;

namespace HLFX
{
    public class SimulationContext
    {
        /// <summary>
        /// 計算時間間隔[s]
        /// 原則60s
        /// 壁,窓などはさらに細かい時間間隔で行う
        /// </summary>
        public long Dt { get; set; }

        /// <summary>
        /// 現日付・時刻
        /// </summary>
        public DateTime Current { get; internal set; }
    }
}

```

```

/// <summary>
/// 前時点日付・時刻
/// </summary>
public DateTime Before { get; internal set; }

/// <summary>
/// 現時刻(時刻・分を4桁の整数化したもの)
/// </summary>
public int CurrentTime
{
    get
    {
        return Current.Hour * 100 + Current.Minute;
    }
}

/// <summary>
/// 前時点の時刻(時刻・分を4桁の整数化したもの)
/// </summary>
public int BeforeTime
{
    get
    {
        return Before.Hour * 100 + Before.Minute;
    }
}

/// <summary>
/// 期の定義 0:暖房期, 1:冷房期, 2:中間期
/// </summary>
public int[] Term { get; set; }

/// <summary>
/// 曜日の定義 0:平日, 1:休日
/// </summary>
public int[] DayoftheWeek { get; set; }

/// <summary>
/// 現在の期を返す
/// 0:暖房期, 1:冷房期, 2:中間期
/// </summary>
public int CurrentTerm
{
    get
    {
        DateTime date = new DateTime(2015, 1, 1);
        TimeSpan ts = Current.Date - date;
        return Term[ts.Days];
    }
}

/// <summary>
/// 前時点の期を返す
/// 0:暖房期, 1:冷房期, 2:中間期
/// </summary>
public int BeforeTerm
{
    get
    {
        DateTime date = new DateTime(2015, 1, 1);
        TimeSpan ts = Before.Date - date;
        return Term[ts.Days];
    }
}

/// <summary>
/// 経過時間[秒]

```

```

/// </summary>
public long Elapsed { get; internal set; }
public IList<IBlock> Blocks { get; internal set; }

/// <summary>
/// 現在の曜日を返す
/// 0(日), 6(土), 7(祝)は1を, それ以外は0を返す
/// </summary>
public int CurrentWeek
{
    get
    {
        DateTime date = new DateTime(2015, 1, 1);
        TimeSpan ts = Current.Date - date;
        if (1 <= DayoftheWeek[ts.Days] &&
            DayoftheWeek[ts.Days] <= 5)
            return 0;
        else
            return 1;
    }
}

/// <summary>
/// 前時点の曜日を返す
/// 0(日), 6(土), 7(祝)は1を, それ以外は0を返す
/// </summary>
public int BeforeWeek
{
    get
    {
        DateTime date = new DateTime(2015, 1, 1);
        TimeSpan ts = Before.Date - date;
        if (1 <= DayoftheWeek[ts.Days] &&
            DayoftheWeek[ts.Days] <= 5)
            return 0;
        else
            return 1;
    }
}

public void SetTerm(DateTime heatingtermstartday,
    DateTime heatingtermendday, DateTime
    coolingtermstartday, DateTime coolingtermendday)
{
    Term = new int[365];
    DayoftheWeek = new int[365];

    for (int i = 0; i < 365; i++)
    {
        Term[i] = 2; //中間期で初期化
        DayoftheWeek[i] = 0; //平日で初期化
    }

    DateTime sd = new DateTime(2015, 1, 1); //1月1日
    DateTime ed = new DateTime(2014, 12, 31); //前年)12月31日

    ///
    ///暖房期の設定
    ///
    if (heatingtermstartday < ed) //暖房
    期間が年をまたぐ場合
    {
        TimeSpan a = heatingtermendday - sd; //1月1日
        から暖房終了日まで
        for (int i = 0; i <= a.Days; i++)

```

```

        Term[i] = 0;

        TimeSpan b = ed - heatingtermstartday; //暖房開
        始日(前年)から12月31日まで
        for (int i = 0; i <= b.Days; i++)
            Term[364-i] = 0;
    }
    else
    {
        TimeSpan a = heatingtermstartday - sd;
        TimeSpan b = heatingtermendday - sd;

        for (int i = a.Days; i <= b.Days; i++)
            Term[i] = 0;
    }

    ///
    ///冷房期の設定
    ///
    if (coolingtermstartday < ed) //冷房
    期間が年をまたぐ場合
    {
        TimeSpan a = coolingtermendday - sd; //1月1日
        から冷房終了日まで
        for (int i = 0; i <= a.Days; i++)
            Term[i] = 1;

        TimeSpan b = ed - coolingtermstartday; //暖房開
        始日(前年)から12月31日まで
        for (int i = 0; i <= b.Days; i++)
            Term[364 - i] = 1;
    }
    else
    {
        TimeSpan a = coolingtermstartday - sd;
        TimeSpan b = coolingtermendday - sd;

        for (int i = a.Days; i <= b.Days; i++)
            Term[i] = 1;
    }

    ///
    ///曜日の設定
    ///1月1日を日曜とする
    ///
    int ii = 0;
    for (int i = 0; i < 365; i++)
    {
        DayoftheWeek[i] = ii++; //代入後にiiをインクリ
        メント
        if (ii == 7) ii = 0; //7になったら0に戻す
    }

    ///祝日
    List<DateTime> DateList = new List<DateTime>();

    DateList.Add(new DateTime(2015, 1, 1));
    DateList.Add(new DateTime(2015, 1, 15));
    DateList.Add(new DateTime(2015, 2, 11));
    DateList.Add(new DateTime(2015, 3, 21));
    DateList.Add(new DateTime(2015, 4, 29));
    DateList.Add(new DateTime(2015, 5, 3));
    DateList.Add(new DateTime(2015, 5, 4));
    DateList.Add(new DateTime(2015, 5, 5));
    DateList.Add(new DateTime(2015, 7, 20));

```



```

DateList.Add(new DateTime(2015, 9, 15));
DateList.Add(new DateTime(2015, 9, 23));
DateList.Add(new DateTime(2015, 10, 10));
DateList.Add(new DateTime(2015, 11, 3));
DateList.Add(new DateTime(2015, 11, 23));
DateList.Add(new DateTime(2015, 12, 23));

for (int n = 0; n < DateList.Count; n++)
{
    TimeSpan a = DateList[n] - sd;
    DayoftheWeek[a.Days] = 7;
}

}

/// <summary>
/// 日付を1/1からの通算日に変換する
/// </summary>
public static int GetTotalDay(DateTime today)
{
    DateTime nd = new DateTime(2015, 1, 1);
    TimeSpan a = today - nd;
    return a.Days + 1;
}
}
}

```

1.4 IBlock.cs

```

namespace HLFX.Block
{
    public interface IBlock
    {
        bool IsComposite { get; }
        string Name { get; set; }
        IConnector[] GetConnectors();

        /// <summary>
        /// オブジェクトごとの計算時間間隔 単位注意[ms]
        /// </summary>
        /// <param name="context"></param>
        long Dt { get; set; }

        /// <summary>
        /// 初期化 (一度だけ)
        /// </summary>
        /// <param name="context"></param>
        void Init(SimulationContext context);

        /// <summary>
        /// 初期化 (反復前に毎回)
        /// </summary>
        /// <param name="context"></param>
        void IterationInit(SimulationContext context);

        /// <summary>
        /// 計算実行
        /// </summary>
        /// <param name="context"></param>
        void Run(SimulationContext context);

        /// <summary>

```

```

/// 後処理 (反復後に毎回)
/// </summary>
/// <param name="context"></param>
void Commit(SimulationContext context);

}
}

```

1.5 IConnector.cs

```

namespace HLFX.Block
{
    public interface IConnector
    {
        string Name { get; set; }

        void AddHeat(double Qc);
        void AddMoisture(double dJ, IConnector src, IConnector
kernel);
        void AddMoisture(double dJ);
        double GetHeat();
        double GetMoisture();
        void Run(SimulationContext context);
        void Commit(SimulationContext context);
    }
}

```

1.6 BaseBlock.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace HLFX.Block
{
    public abstract class BaseBlock : IBlock, IConnector
    {
        public string Name { get; set; }

        /// <summary>
        /// オブジェクトごとの計算時間間隔 単位[ms]
        /// </summary>
        public virtual long Dt { get; set; }

        public virtual bool IsComposite { get { return
false; } }

        public virtual IConnector[] GetConnectors()
        {
            return null;
        }

        /// <summary>
        /// 内部熱量[J]
        /// </summary>
        protected double U;

        /// <summary>

```

```

/// 熱移動量[J]
/// </summary>
protected double dU_tick;

/// <summary>
/// 内部水分量[kg]
/// </summary>
protected double J;

/// <summary>
/// 水分移動量[kg]
/// </summary>
protected double dJ_tick;

/// <summary>
/// 熱移動
/// </summary>
/// <param name="dU">熱移動量[J]</param>
public virtual void AddHeat(double dU, IConnector src,
IConnector kernel)
{

public virtual void AddHeat(double dU)
{
    dU_tick += dU;
}

public virtual double GetHeat()
{
    return dU_tick;
}

/// <summary>
/// 水分移動
/// </summary>
/// <param name="dJ">水分伝達量 (単位[kg]) </param>
public virtual void AddMoisture(double dJ, IConnector
src, IConnector kernel)
{
    dJ_tick += dJ;
    System.Diagnostics.Debug.Assert(dJ != -0.0002);
}

public virtual void AddMoisture(double dJ)
{
    dJ_tick += dJ;
}

public virtual double GetMoisture()
{
    return dJ_tick;
}

public virtual void Init(SimulationContext context)
{
}

public virtual void IterationInit(SimulationContext
context)
{
}

public virtual void Run(SimulationContext context)
{
}

public virtual void Commit(SimulationContext context)

```

```

{
    var children = GetConnectors();
    if (children != null)
    {
        foreach (var child in children)
        {
            child.Commit(context);
        }
    }

    U += dU_tick;
    J += dJ_tick;

    dU_tick = 0.0;
    dJ_tick = 0.0;
}

/// <summary>
/// ブロックごとのDtを求める(60sの約数, 単位ms)
/// </summary>
/// <param name="dt">ブロックごとの収束条件を満たす
dt(単位ms) </param>
public long CalcDt(long dt)
{
    long[] tmpdt = new long[5];
    tmpdt[0] = 60000; //60s × 1回
    tmpdt[1] = 30000; //30s × 2回
    tmpdt[2] = 20000; //20s × 3回
    tmpdt[3] = 15000; //15s × 4回
    tmpdt[4] = 10000; //10s × 6回

    //これ以降は[0]~[4]を10および100で除した値とする

    if (dt >= tmpdt[4])
    {
        for (int n = 0; n < 5; n++)
        {
            if (dt >= tmpdt[n])
                return tmpdt[n];
        }
    }
    else if (dt >= tmpdt[4] / 10)
    {
        for (int n = 0; n < 5; n++)
        {
            if (dt >= (tmpdt[n] / 10))
                return tmpdt[n] / 10;
        }
    }
    else if (dt >= tmpdt[4] / 100)
    {
        for (int n = 0; n < 5; n++)
        {
            if (dt >= (tmpdt[n] / 100))
                return tmpdt[n] / 100;
        }
    }
    else if (dt >= tmpdt[4] / 1000)
    {
        for (int n = 0; n < 4; n++)
        {
            if (dt >= (tmpdt[n] / 1000))
                return tmpdt[n] / 1000;
        }
    }

    return 0;
}

```

```

}
}
}

```

1.7 ISpaceBlock.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 空間
    /// </summary>
    public interface ISpaceBlock : IConnector
    {
        /// <summary>
        /// 温度[°C]
        /// </summary>
        double Temp { get; }

        /// <summary>
        /// MRT[°C]
        /// </summary>
        double MRT { get; set; }

        /// <summary>
        /// 絶対湿度[kg/kg']
        /// </summary>
        double Humidity { get; }

        /// <summary>
        /// 窓から取得する直達日射(放射成分)[W]
        /// 単位注意(窓面積を乗じた和)
        /// </summary>
        double DirectSolarRadiationTotal { get; set; }

        /// <summary>
        /// 窓から取得する直達日射(対流成分)[W]
        /// 単位注意(窓面積を乗じた和)
        /// </summary>
        double DirectSolarConvectionTotal { get; set; }

        double SkySolarRadiationTotal { get; set; }
        double SkySolarConvectionTotal { get; set; }
        double ReflectSolarRadiationTotal { get; set; }
        double ReflectSolarConvectionTotal { get; set; }

        /// <summary>
        /// 壁などの面積当りに配分される内部発熱(放射成分)を
        /// 返す [W/m2]
        /// </summary>
        double SensibleHeatGenerationRadiationToWall { get; }

        /// <summary>
        /// 床面積当りに配分される日射量(放射成分)を返す
        /// [W/m2]
        /// </summary>

```

```

        double SolarRadiationToFloor { get; }

        /// <summary>
        /// 壁など(床以外)の床面積当りに配分される日射量(放
        /// 射成分)を返す [W/m2]
        /// </summary>
        double SolarRadiationToWall { get; }

        /// <summary>
        /// 面の角度を受け取り,当該面に入射する日射量,夜間放射
        /// 量を返す
        /// </summary>
        void GetSolar(double costheta, double facecosh, out
        double directsolar, out double skysolar, out double
        reflectsolar, out double longwave);

        /// <summary>
        /// 面と太陽方向のなす角のcosを返す
        /// </summary>
        double GetCosTheta(double facewz, double faceww,
        double facews);

        /// <summary>
        /// 壁のSAT温度(日射吸収あり)
        /// </summary>;
        double GetSAT(double solarabsorptionrate, double
        longwaveemissionrate, double alphas, double costheta,
        double facecosh);

        /// <summary>
        /// 窓のSAT温度(日射吸収なし)
        /// </summary>
        double GetSAT(double longwaveemissionrate, double
        alphas, double costheta, double facecosh);

        void SetFaceConnector(WallFaceConnector
        connectingface, int facetype);

        void SetVentilation(VentilationHeatTransferBlock
        ventilation);

        void AddVent(double qv, double jv);
    }
}

```

1.8 ILayerBlock.cs

```

using System;
namespace HLFX.Block
{
    public interface ILayerBlock : IConnector
    {
        /// <summary>
        /// 隣接するオブジェクト(空間または外部)
        /// </summary>
        ISpaceBlock Space { get; }

        /// <summary>
        /// 隣接するオブジェクト表面(壁または窓の表面)
        /// </summary>
        ISurface Surface { get; }
    }
}

```

```

/// <summary>
/// 表面積[m^2]
/// </summary>
double Area { get; }

/// <summary>
/// 対流熱伝達率[W/m2K]
/// </summary>
double AlphaC { get; }

/// <summary>
/// 放射熱伝達率[W/m2K]
/// </summary>
double AlphaR { get; }

/// <summary>
/// 日射吸収率[-]
/// </summary>
double SolarAbsorptionRate { get; }

/// <summary>
/// 長波長放射率[-]
/// </summary>
double LongWaveEmissionRate { get; }

/// <summary>
/// 水分伝達率[kg/m2s(kg/kg' )]
/// </summary>
double MoistureTransferRate { get; }

/// <summary>
/// 空気温度[°C]
/// </summary>
double SpaceTemp { get; }

/// <summary>
/// 空気絶対湿度 [kg/kg']
/// </summary>
double SpaceHumidity { get; }

/// <summary>
/// 壁窓表面温度[°C]
/// </summary>
double SurfaceTemp { get; }

/// <summary>
/// 壁窓表面絶対湿度 [kg/kg']
/// </summary>
double SurfaceHumidity { get; }
}

public interface ISurface
{
/// <summary>
/// 壁・窓ブロックのA側表面温度
/// </summary>
double TempA { get; }

/// <summary>
/// 壁・窓ブロックのB側表面温度
/// </summary>
double TempB { get; }

/// <summary>
/// 壁・窓ブロックのA側表面絶対湿度
/// </summary>

```

```

double HumidityA { get; }

/// <summary>
/// 壁・窓ブロックのB側表面温度
/// </summary>
double HumidityB { get; }

/// <summary>
/// A側の熱流を与える(空気→表面の向きを正)
/// </summary>
void AddHeatA();

/// <summary>
/// B側の熱流を与える(空気→表面の向きを正)
/// </summary>
void AddHeatB();

/// <summary>
/// A側の水分流を与える(空気→表面の向きを正)
/// </summary>
void AddMoistureA();

/// <summary>
/// B側の水分流を与える(空気→表面の向きを正)
/// </summary>
void AddMoistureB();

double Costheta { get; set; }
}
}

```

1.9 RoomBlock.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
public class RoomBlock : BaseBlock, ISpaceBlock
{
private const double rho_air = 1.2;
private const double c_air = 1005.0;

public struct ConnectingFace
{
public int FaceType; //0:床以外, 1:床
public WallFaceConnector Face;

public ConnectingFace(WallFaceConnector face, int
facetyp)
{
Face = face;
FaceType = facetyp;
}
}

/// <summary>
/// 接続する表面の数

```

```

/// </summary>
public int ConnectingFaceNum { get; set; }

/// <summary>
/// 接続する表面
/// </summary>
private List<ConnectingFace> ListConnectingFace = new
List<ConnectingFace>();

/// <summary>
/// 接続する換気経路(自分が下流のものに限る)
/// </summary>
private List<VentilationHeatTransferBlock>
ListVentilation = new
List<VentilationHeatTransferBlock>();

/// <summary>
/// 外部環境への参照
/// </summary>
public OutsideBlock Outside;

/// <summary>
/// 設置する発熱体
/// </summary>
public List<HeatGenerator> ListHeatGenerator = new
List<HeatGenerator>();

/// <summary>
/// 在室者(複数人)
/// </summary>
public PersonBlock Person;

/// <summary>
/// 設置するエアコン 当面1台だけとする
/// </summary>
public AirConditionerBlock AC;

/// <summary>
/// 気積[m3]
/// </summary>
public double Volume { get; set; }

/// <summary>
/// 換気回数(窓CLOSE)
/// </summary>
public double VentNumClose { get; set; }

/// <summary>
/// 換気回数(窓OPEN)
/// </summary>
public double VentNumOpen { get; set; }

/// <summary>
/// 顕熱容量[J/K]
/// </summary>
public double Cs { get; set; }

/// <summary>
/// 潜熱容量[kg/(kg/kg')]
/// </summary>
public double Ci { get; set; }

/// <summary>
/// 表面積合計[m2]
/// </summary>
public double TotalArea { get; set; }

```

```

/// <summary>
/// 床として定義された面の面積合計[m2]
/// </summary>
public double TotalFloorArea { get; set; }

/// <summary>
/// 在室スケジュール
/// </summary>
public ScheduleSeries[,] OccupySchedule;

#region 状態

/// <summary>
/// 室空気温度[°C]
/// </summary>
public double Temp { get; set; }

/// <summary>
/// 室絶対湿度 [kg/kg']
/// </summary>
public double Humidity { get; set; }

/// <summary>
/// 室相対湿度 [-]
/// </summary>
public double RHumidity { get; set; }

/// <summary>
/// 平均表面温度 [°C]
/// </summary>
public double MRT { get; set; }

/// <summary>
/// 換気流入熱量の和[W]
/// </summary>
public double Qvent { get; set; }

/// <summary>
/// 換気流入水分量の和[kg/s]
/// </summary>
public double Jvent { get; set; }

/// <summary>
/// エアコン処理熱量の和[W]
/// </summary>
public double Qac { get; internal set; }

/// <summary>
/// エアコン処理水分量の和[kg/s]
/// </summary>
public double Jac { get; internal set; }

/// <summary>
/// 窓から取得する直達日射(放射成分) [W]
/// 単位注意(窓面積を乗じた和)
/// </summary>
public double DirectSolarRadiationTotal { get; set; }

/// <summary>
/// 窓から取得する直達日射(対流成分) [W]
/// 単位注意(窓面積を乗じた和)
/// </summary>
public double DirectSolarConvectionTotal { get; set; }

public double SkySolarRadiationTotal { get; set; }

```

```

public double SkySolarConvectionTotal { get; set; }

public double ReflectSolarRadiationTotal { get; set; }

public double ReflectSolarConvectionTotal { get;
set; }

/// <summary>
/// 内部発熱量(顕熱, 対流成分)の和[W]
/// </summary>
public double SensibleHeatGenerationTotalConvection
{ get; internal set; }

/// <summary>
/// 内部発熱量(顕熱, 放射成分)の和[W]
/// </summary>
public double SensibleHeatGenerationTotalRadiation
{ get; internal set; }

/// <summary>
/// 内部発熱量(潜熱)の和[kg/s]
/// </summary>
public double VaporGenerationTotal { get; internal
set; }

/// <summary>
/// 壁などの面積当たりに配分される内部発熱(放射成分)を
返す [W/m2]
/// </summary>
public double SensibleHeatGenerationRadiationToWall
{
    get
    {
        if (TotalArea > 0.0)
            return SensibleHeatGenerationTotalRadiation /
                TotalArea;
        else
            return 0.0;
    }
}

/// <summary>
/// 床以外への単位面積当たりの日射配分量(放射成分)を返
す[W/m2]
/// </summary>
public double SolarRadiationToWall
{
    get {
        if (TotalFloorArea > 0.0) //床がある
        {
            if (TotalArea > TotalFloorArea) //壁もある
                return 0.5 * (DirectSolarRadiationTotal +
                    SkySolarRadiationTotal +
                    ReflectSolarRadiationTotal) / (TotalArea -
                        TotalFloorArea);
            else //壁がない(床だけ)
                return 0.0;
        }
        else //床が定義されていない場合
        {
            if (TotalArea > 0.0)
                return (DirectSolarRadiationTotal +
                    SkySolarRadiationTotal +
                    ReflectSolarRadiationTotal) / TotalArea;
            else
                return 0.0;
        }
    }
}

```

```

}
}
}

/// <summary>
/// 床への単位面積当たりの日射配分量(放射成分)を返す
[W/m2]
/// </summary>
public double SolarRadiationToFloor
{
    get
    {
        if (TotalFloorArea > 0.0)
            return 0.5 * (DirectSolarRadiationTotal +
                SkySolarRadiationTotal +
                ReflectSolarRadiationTotal) / TotalFloorArea;
        else
            return 0.0;
    }
}

public int WindowOpen { get; set; }

public int ACONOFF { get; set; }

#endregion

/// <summary>
/// 室の計算時間間隔[s]
/// </summary>
public override long Dt { get; set; }

/// <summary>
/// 1minにおける室の計算回数
/// </summary>
private long IterationNum;

public void SetFaceConnector(WallFaceConnector
connectingface, int facetype)
{
    ConnectingFace face = new
    ConnectingFace(connectingface, facetype);
    ListConnectingFace.Add(face);
}

public void
SetVentilation(VentilationHeatTransferBlock
ventilation)
{
    ListVentilation.Add(ventilation);
}

public void SetHeatGenerator(HeatGenerator
heatgenerator)
{
    ListHeatGenerator.Add(heatgenerator);
}

public void SetPerson(PersonBlock person)
{
    Person = person;
}

public void SetAirConditioner(AirConditionerBlock ac)
{
    AC = ac;
}

```

```

}

/// <summary>
/// 面の角度を受け取り, 当該面に入射する日射量, 夜間放射
/// 量を返す
/// </summary>
public void GetSolar(double costheta, double facecosh,
out double directsolar, out double skysolar, out
double reflectsolar, out double longwave)
{
    directsolar = 0.0;
    skysolar = 0.0;
    reflectsolar = 0.0;
    longwave = 0.0;
}

/// <summary>
/// 面の角度を受け取り, 太陽方向となす角のcosを返す
/// </summary>
/// <param name="facewz">面のcos(傾斜角)</param>
/// <param name="faceww">面のsin(傾斜角) × sin(方位
/// 角)</param>
/// <param name="faceww">面のsin(傾斜角) × cos(方位
/// 角)</param>
public double GetCosTheta(double facewz, double
faceww, double facews)
{
    return 0.0;
}

public override void Init(SimulationContext context)
{
    /// 顕熱容量
    if (Cs <= 0.0) Cs = c_air * rho_air * Volume;
    if (Cl <= 0.0) Cl = rho_air * Volume;

    Qvent = 0.0;
    Jvent = 0.0;
    Qac = 0.0;
    Jac = 0.0;
    DirectSolarRadiationTotal = 0.0;
    DirectSolarConvectionTotal = 0.0;
    SkySolarRadiationTotal = 0.0;
    SkySolarConvectionTotal = 0.0;
    ReflectSolarRadiationTotal = 0.0;
    ReflectSolarConvectionTotal = 0.0;
    SensibleHeatGenerationTotalConvection = 0.0;
    SensibleHeatGenerationTotalRadiation = 0.0;
    VaporGenerationTotal = 0.0;

    ///
    /// 対流・総合熱伝達相手先の集計
    ///
    double alphaca = 0.0;    ///  $\sum \alpha cA$ 
    foreach (ConnectingFace face in ListConnectingFace)
    {
        alphaca += face.Face.Area * face.Face.AlphaC;
        TotalArea += face.Face.Area;

        if (face.FaceType == 1)
            TotalFloorArea += face.Face.Area;
    }

    ///
    /// 換気相手先(自分が下流のもの)の集計

```

```

///
double crhov = 0.0;    ///  $\sum c\rho V$ 
double rhov = 0.0;    ///  $\sum \rho V$ 

crhov += c_air * rho_air * Math.Max(VentNumClose,
VentNumOpen) * Volume / 3600.0;    /// 自然換気分
rhov += rho_air * Math.Max(VentNumClose, VentNumOpen)
* Volume / 3600.0;    /// 自然換気分

foreach (VentilationHeatTransferBlock ventilation in
ListVentilation)
{
    crhov += c_air * rho_air * ventilation.AirVolume;
    rhov += rho_air * ventilation.AirVolume;
}

double t1 = 60.0;    /// 熱の式から求まる時間間隔 仮
定値(10s)
double t2 = 60.0;    /// 湿気の式から求まる時間間隔 仮
定値(10s)
if (alphaca + crhov > 0.0)
    t1 = Cs / (alphaca + crhov);

if (rhov > 0.0)
    t2 = Cl / rhov;

double t;
if (t1 > t2) t = t2;
else t = t1;

Dt = (long)(t * 1000.0);    /// [ms] 単位で保持する

if (Dt == 0)
{
    System.Environment.Exit(1);
}

Dt = base.CalcDt(Dt);
IterationNum = 60000 / Dt;    /// 1minの反復回数
}

/// <summary>
/// 換気熱・水分移動量を受け取る(換気扇によるもののみ)
/// <param name="qv">換気熱移動量[W]</param>
/// <param name="jv">換気水分移動量[kg/s]</param>
/// </summary>
public void AddVent(double qv, double jv)
{
    Qvent += qv;
    Jvent += jv;
}

/// <summary>
/// エアコンの処理熱量を受け取る
/// </summary>
/// <param name="qac">エアコン処理顕熱[W] 冷房が正
</param>
/// <param name="jac">エアコン処理水分量[kg/s] 除湿が
正</param>
public void AddAC(double qac, double jac)
{
    Qac += qac;
    Jac += jac;
}

/// <summary>

```

```

/// 反復前の毎回の初期化
/// </summary>
/// <param name="context"></param>
public override void IterationInit(SimulationContext
context)
{
    ///
    /// 内部発熱量の総和を求める[W]
    ///
    double qgenc = 0.0;
    double qgenr = 0.0;
    double jgen = 0.0;    ///[kg/s]
    for (int i = 0; i < ListHeatGenerator.Count; i++)
    {
        ListHeatGenerator[i].GetHeat(context, out qgenc,
        out qgenr, out jgen);

        SensibleHeatGenerationTotalConvection += qgenc;
        SensibleHeatGenerationTotalRadiation += qgenr;
        VaporGenerationTotal += jgen;
    }

    ///
    ///在室者
    ///
    if (Person != null)
    {
        Person.GetHeat(context, out qgenc, out qgenr, out
        jgen);
        SensibleHeatGenerationTotalConvection += qgenc;
        SensibleHeatGenerationTotalRadiation += qgenr;
        VaporGenerationTotal += jgen;
    }

    ///
    ///エアコン判定
    ///
    if (AC != null)
    {
        if (AC.WorkType == 0)    ///スケジュールどおり
        {
            AC.SetOperation(context);
            if (AC.currentOperation)
                this.ACONOFF = 1;
            else
                this.ACONOFF = 0;
        }
        else if (AC.WorkType == 1)    ///スケジュール+室温
        参照
        {
            AC.SetOperation(context, this.Temp);
            if (AC.currentOperation)
                this.ACONOFF = 1;
            else
                this.ACONOFF = 0;
        }
        else if (AC.WorkType == 2)    ///人体判定結果
        {
            if (Person != null)
            {
                int windowopen_new = 0;
                int aconoff_new = 0;
                Person.ACONOFF(context, Temp, Outside.Temp,
                WindowOpen, ACONOFF,
                out windowopen_new, out aconoff_new);

                WindowOpen = windowopen_new;
                ACONOFF = aconoff_new;
            }
        }
    }
}

```

```

    }
    AC.SetOperation(this.ACONOFF);
}
}

public override void Run(SimulationContext context)
{
    double qc = 0.0;    ///対流熱伝達量[W]
    foreach (ConnectingFace face in ListConnectingFace)
    {
        qc += face.Face.GetHeat() * face.Face.Area;
        ///GetHeatはスペースから面に移動する量が正[W/m2]
    }
    double qs = DirectSolarConvectionTotal +
    SkySolarConvectionTotal +
    ReflectSolarConvectionTotal;    ///日射の対流成分[W]
    double qg = SensibleHeatGenerationTotalConvection;

    ///自然換気(窓開け判定結果を反映)
    double qnv = 0.0;
    double jnv = 0.0;
    double ventnum = 0.0;
    if (WindowOpen == 0)
        ventnum = VentNumClose;
    else
        ventnum = VentNumOpen;

    qnv += 1005.0 * rho_air * ventnum * Volume / 3600.0 *
    (Outside.Temp - this.Temp);
    jnv += rho_air * ventnum * Volume / 3600.0 *
    (Outside.Humidity - this.Humidity);
    Qvent += qnv;
    Jvent += jnv;

    double TtmpOld = Temp;    ///暫定温度
    double TtmpNew = Temp;

    for (long n = 0; n < IterationNum; n++)    ///この室
    の反復
    {
        TtmpNew = TtmpOld + (qc + Qvent + qs + qg - Qac) *
        (Dt / 1000.0) / Cs;    ///Dtは[ms]なので注意
        TtmpOld = TtmpNew;
    }

    double jc = 0.0;    ///表面水分伝達量

    double XtmpOld = Humidity;    ///暫定絶対湿度
    double XtmpNew = Humidity;    ///暫定絶対湿度

    double jgen = VaporGenerationTotal;    ///内部発熱(潜
    熱) [kg/s]

    for (long n = 0; n < IterationNum; n++)
    {
        XtmpNew = XtmpOld + (jc + Jvent + jgen - Jac) * (Dt
        / 1000.0) / Cl;
        XtmpOld = XtmpNew;
    }

    ///
    ///反復終了
    ///
}

```



```

Temp = TtmpNew;
Humidity = XtmpNew;
RHumidity = Utility.Util.CalcHumidity(Temp, Humidity,
3);

///相対湿度が100%超となったときの処理(100%まで下げる)
if (RHumidity > 1.0)
{
Humidity = Utility.Util.CalcHumidity(Temp, 1.0, 1);
///相湿100%となる絶対湿度
RHumidity = 1.0;
}
}

```

```

public override void Commit(SimulationContext context)
{

```

```

Qvent = 0.0;
Jvent = 0.0;
Qac = 0.0;
Jac = 0.0;
SensibleHeatGenerationTotalRadiation = 0.0;
SensibleHeatGenerationTotalConvection = 0.0;
VaporGenerationTotal = 0.0;

```

```

DirectSolarRadiationTotal = 0.0;
DirectSolarConvectionTotal = 0.0;
SkySolarRadiationTotal = 0.0;
SkySolarConvectionTotal = 0.0;
ReflectSolarRadiationTotal = 0.0;
ReflectSolarConvectionTotal = 0.0;

```

```

///
///表面のMRT( $\sum \theta_j A_j$ )の計算 次時刻の放射熱伝達計算に
用いる
///

```

```

MRT = 0.0;
for ( int n=0 ; n < ListConnectingFace.Count; n++)
{
double temp =
ListConnectingFace[n].Face.SurfaceTemp;

MRT += temp * ListConnectingFace[n].Face.Area;
}
MRT /= TotalArea;

```

```

base.Commit(context);
}

```

```

/// <summary>
/// ダミー
/// </summary>
/// <param name="facewz"></param>
/// <param name="faceww"></param>
/// <param name="facews"></param>
/// <returns></returns>
public double GetCosTheta(double facewz, double faceww,
double facews)
{ return 0.0; }

```

```

public double GetSAT(double solarabsorptionrate,
double longwaveemissionrate, double alphas, double
costheta, double facecosh)
{ return 0.0; }

```

```

public double GetSAT(double longwaveemissionrate,

```

```

double alphas, double costheta, double facecosh)
{ return 0.0; }
}
}

```

1.10 WallBlock.cs

```

using HAFX.Utility;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace HAFX.Block

```

```

{
/// <summary>
/// 壁
/// </summary>
public class WallBlock : BaseBlock, ISurface

```

```

{
/// <summary>
/// 面積[m2]
/// </summary>
public double Area { get; set; }

```

```

/// <summary>
/// A側スペースにとつての床フラグ(1なら床)
/// </summary>
public int FloorofSpaceA { get; set; }

```

```

/// <summary>
/// 材料厚さ[m]
/// </summary>
public double[] Thickness { get; }

```

```

/// <summary>
/// 材料
/// </summary>
public Material[] Material { get; }

```

```

/// <summary>
/// 層構成の向き 1:層構成のI側が壁のA側, 2:逆
/// </summary>
public int MatDirection { get; }

```

```

/// <summary>
/// 表面A
/// </summary>
public WallFaceConnector FaceA { get; private set; }

```

```

/// <summary>
/// 表面B
/// </summary>
public WallFaceConnector FaceB { get; private set; }

```

```

public double Costheta { get; set; }

```

```

private int NodeNumber;

```

```

private double[] q_WL;
private double[] J_WL;
private double[] gamma;

```

```

private double[] gamma_x;
private double[] d;
private double[] T;           //節点温度
private double[] Tnew;       //新節点温度
private double[] X;          //節点湿度
private double[] Xnew;       //新節点湿度

private double[] NodeR;      //節点間熱抵抗 節点
// [0]と節点[1]の間を[0]とする
private double[] NodeC;      //節点熱容量

/// <summary>
/// 節点温度 I→J
/// </summary>
public double[] Temp { get { return T; } }

/// <summary>
/// 節点湿度 I→J
/// </summary>
public double[] Humidity { get { return X; } }

/// <summary>
/// A側表面温度[°C]
/// </summary>
public double TempA
{
    get
    {
        if (MatDirection == 1)
            return T[0];
        else
            return T[NodeNumber - 1];
    }
}

/// <summary>
/// B側表面温度[°C]
/// </summary>
public double TempB
{
    get
    {
        if (MatDirection == 1)
            return T[NodeNumber - 1];
        else
            return T[0];
    }
}

/// <summary>
/// A側表面絶対湿度[kg/kg']
/// </summary>
public double HumidityA
{
    get { return X[0]; }
}

/// <summary>
/// B側表面絶対湿度[kg/kg']
/// </summary>
public double HumidityB
{
    get { return X[1]; }
}

/// <summary>
/// A側表面熱取得量[J]

```

```

/// </summary>
public double dU_tick_A { get; private set; }

/// <summary>
/// B側表面熱取得量[J]
/// </summary>
public double dU_tick_B { get; private set; }

/// <summary>
/// 壁の計算時間間隔[s]
/// </summary>
public override long Dt { get; set; }

/// <summary>
/// 1minにおける壁の計算回数
/// </summary>
private long IterationNum;

public override bool IsComposite
{
    get
    {
        return true;
    }
}

public override IConnector[] GetConnectors()
{
    return new IConnector[] { FaceA, FaceB };
}

/// <summary>
/// コンストラクタ
/// </summary>
/// <param name="spaceA">A側居室</param>
/// <param name="spaceB">B側居室または外部</param>
/// <param name="area">面積[m^2]</param>
/// <param name="floorofspaceA">A側居室にとっての床フ
ラグ</param>
/// <param name="materialList">材料の配列</param>
/// <param name="thicknessList">材料厚さ[m]の配列
</param>
/// <param name="matdirection">層構成の向き(1:層構成I
側が壁のA側, 2:逆)</param>
/// <param name="alphacA">A側対流熱伝達率
[W/m2K]</param>
/// <param name="alphanA">A側放射熱伝達率
[W/m2K]</param>
/// <param name="solarabsorptionrateA">A側日射吸収率
[-]</param>
/// <param name="longwaveemissionrateA">A側長波放射率
[-]</param>
/// <param name="alphacB">B側対流熱伝達率
[W/m2K]</param>
/// <param name="alphanB">B側放射熱伝達率
[W/m2K]</param>
/// <param name="solarabsorptionrateB">B側日射吸収率
[-]</param>
/// <param name="longwaveemissionrateB">B側長波放射率
[-]</param>
/// <param name="faceh">面傾斜角[deg]</param>
/// <param name="facea">面方位角[deg]</param>
public WallBlock(string name, ISpaceBlock spaceA,
ISpaceBlock spaceB, double area, int floorofspaceA,
Material[] materialList, double[] thicknessList, int
matdirection,
double alphacA, double alphanA, double

```

```

solarabsorptionrateA, double longwaveemissionrateA,
double alphacB, double alphasB, double
solarabsorptionrateB, double longwaveemissionrateB,
double faceh, double facea)
: base()
{
this.Name = name;
this.Area = area;
this.FloorofSpaceA = floorofspaceA;
this.Thickness = thicknessList;
this.Material = materialList;
this.MatDirection = matdirection;
this.FaceA = new WallFaceConnector(name+"A", spaceA,
this, 1, 0, area, alphacA, alphasA,
solarabsorptionrateA, longwaveemissionrateA, faceh,
facea);
this.FaceB = new WallFaceConnector(name+"B", spaceB,
this, 2, 6, area, alphacB, alphasB,
solarabsorptionrateB, longwaveemissionrateB, faceh,
facea);

//spaceAに登録
if (spaceA is RoomBlock)
{
spaceA.SetFaceConnector(this.FaceA, floorofspaceA);
}

//spaceBに登録
if (spaceB is RoomBlock)
{
spaceB.SetFaceConnector(this.FaceB, 0);
}

//作業領域作成
int matnum = materialList.Length; //材料数
NodeNumber = materialList.Length; //原則として節
点数=材料数とする, 端部材料節点は材料表面にあり, 内部
材料節点は材料中央にあるものとする
if (matnum == 1) NodeNumber = 2; //材料数=1の場合
は両側に節点を置く

d = new double[matnum]; //各材料厚さ
for (int i = 0; i < matnum; i++)
{
d[i] = thicknessList[i];
}

double[] lambda = new double[matnum]; //各材料熱伝
導率
double[] cr = new double[matnum]; //各材料容積
比熱
double[] mr = new double[matnum]; //各材料熱抵
抗
double[] mc = new double[matnum]; //各材料熱容
量
for (int i = 0; i < matnum; i++)
{
lambda[i] = materialList[i].熱伝導率; //空気層
の場合は0なので注意
cr[i] = materialList[i].容積比熱;
//[J/m3K]

if (lambda[i] > 0.0)
mr[i] = d[i] / lambda[i];
else
mr[i] = materialList[i].空気層熱抵抗;

```

```

mc[i] = cr[i] * d[i];
}

NodeR = new double[NodeNumber - 1];
NodeC = new double[NodeNumber];

if (matnum == 1) //材料数1(節
点数 = 2)
{
NodeR[0] = mr[0];
NodeC[0] = NodeC[1] = 0.5 * mc[0];
}
else if (matnum == 2) //材料数2(節
点数 = 2)
{
NodeR[0] = mr[0] + mr[1];
NodeC[0] = mc[0];
NodeC[1] = mc[1];
}
else //材料数3以上(節
点数 = 材料数)
{
for (int i = 0; i < NodeNumber - 1; i++)
{
if (i == 0) //I
端節点と2番目節点(節点番号0, 1)
NodeR[i] = mr[i] + 0.5 * mr[i + 1];
else if (i == NodeNumber - 2) //J
端の手前節点とJ端節点(節点番号NodeNumber-2,
NodeNumber-1)
NodeR[i] = 0.5 * mr[i] + mr[i + 1];
else //
中間節点
NodeR[i] = 0.5 * mr[i] + 0.5 * mr[i + 1];

NodeC[i] = mc[i];
}
NodeC[NodeNumber - 1] = mc[NodeNumber - 1];
}

//収束条件のチェック
double t = 60.0; // 20s
double dt_tmp = 0.0;

for (int i = 0; i < NodeNumber; i++)
{
if (i == 0)
{
if (MatDirection == 1)
dt_tmp = NodeC[i] / (alphacA + alphasA + 1.0 /
NodeR[i]); //材料I側=壁A側
else
dt_tmp = NodeC[i] / (alphacB + alphasB + 1.0 /
NodeR[i]); //材料I側=壁B側
}
else if (i == NodeNumber - 1)
{
if (MatDirection == 1)
dt_tmp = NodeC[i] / (alphacB + alphasB + 1.0 /
NodeR[i - 1]); //材料J側=壁B側
else
dt_tmp = NodeC[i] / (alphacA + alphasA + 1.0 /
NodeR[i - 1]); //材料J側=壁A側
}
else

```

```

    dt_tmp = NodeC[i] / (1.0 / NodeR[i - 1] + 1.0 /
NodeR[i]);

    if (dt_tmp < t) t = dt_tmp;    ///時間間隔制限が最も厳しいものを保持
}
Dt = (long) (t * 1000.0);    ///[ms]単位で保持する

if (Dt == 0)
{
    System.Environment.Exit(1);
}

Dt = base.CalcDt(Dt);
IterationNum = 60000 / Dt;    ///20sを進めるための反復回数

q_WL = new double[matnum + 1];    /////
J_WL = new double[matnum + 1];    /////
gamma = new double[matnum];    /////
gamma_x = new double[matnum];    /////
T = new double[NodeNumber];    /////節点温度
Tnew = new double[NodeNumber];    /////新節点温度
X = new double[NodeNumber];    /////節点絶対湿度
Xnew = new double[NodeNumber];    /////新節点絶対湿度

double initialtemp = 20.0;
double initialhumidity = 7.354 / 1000.0;
for (int i = 0; i < NodeNumber; i++)
{
    T[i] = initialtemp;
    Tnew[i] = initialtemp;
    X[i] = initialhumidity;
    Xnew[i] = initialhumidity;
}
}

///<summary>
///熱移動
///</summary>
///<param name="dU">熱移動量[J]</param>
public void AddHeatA(double dU, IConnector src,
IConnector kernel)
{
    dU_tick += dU;
}

public override void Run(SimulationContext context)
{
    const double k = 100000.0;
    const double v = 0.0;

    /////熱流・水分流
    double[] TtmpOld = new double[NodeNumber];    /////1min
    進行するまでの(仮)更新後温度
    double[] TtmpNew = new double[NodeNumber];    /////1min
    進行するまでの(仮)更新後温度

    /////I側・J側表面
    double qi = 0.0;    /////i側隣接節点からの熱

```

```

[W/m2]
double qj = 0.0;    /////j側隣接節点からの熱
[W/m2]
double qspci = 0.0;    /////i側スペースからの対流熱流[W/m2]
double qspcj = 0.0;    /////j側スペースからの対流熱流[W/m2]
double gradi = 0.0;    /////i側スペースからの放射熱流[W/m2]
double qradj = 0.0;    /////j側スペースからの放射熱流[W/m2]
double qgeni = 0.0;    /////i側スペースの内部発熱(放射成分)[W/m2]
double qgenj = 0.0;    /////j側スペースの内部発熱(放射成分)[W/m2]
double qsoli = 0.0;    /////i側スペースの日射配分(放射成分)[W/m2]
double qsolj = 0.0;    /////j側スペースの日射配分(放射成分)[W/m2]

for (int i = 0; i < NodeNumber; i++)
    TtmpOld[i] = T[i];

if (MatDirection == 1)    /////材料i側=面A側
{
    qspci = FaceA.GetHeat();    /////先頭材料はA側
    qspcj = FaceB.GetHeat();    /////末尾材料はB側

    if (FaceA.Space is FixedConditionRoomBlock)
    {
    }
    else if (FaceA.Space is RoomBlock)
    {
        gradi = FaceA.AlphaR * (FaceA.Space.MRT -
TtmpOld[0]);    /////先頭材料への放射熱
        qgeni =
FaceA.Space.SensibleHeatGenerationRadiationToWall;

        if (this.FloorofSpaceA == 1)
            qsoli = FaceA.Space.SolarRadiationToFloor;
        else
            qsoli = FaceA.Space.SolarRadiationToWall;
    }

    if (FaceB.Space is FixedConditionRoomBlock)
    {
    }
    else if (FaceB.Space is RoomBlock)
    {
        qradj = FaceB.AlphaR * (FaceB.Space.MRT -
TtmpOld[NodeNumber - 1]);    /////末尾材料への放射熱
        qgenj =
FaceB.Space.SensibleHeatGenerationRadiationToWall;
        qsolj = FaceB.Space.SolarRadiationToWall;
    }
}
else
{
    qspci = FaceB.GetHeat();    /////先頭材料はB側
    qspcj = FaceA.GetHeat();    /////末尾材料はA側

    if (FaceA.Space is FixedConditionRoomBlock)
    {
    }
    else if (FaceA.Space is RoomBlock)
    {
        qradj = FaceA.AlphaR * (FaceA.Space.MRT -

```

```

TtmpOld[NodeNumber - 1]); //末尾材料への放射熱
qgenj =
FaceA.Space.SensibleHeatGenerationRadiationToWall;

if (this.FloorofSpaceA == 1)
    qsolj = FaceA.Space.SolarRadiationToFloor;
else
    qsolj = FaceA.Space.SolarRadiationToWall;
}

if (FaceB.Space is FixedConditionRoomBlock)
{
}
else if (FaceB.Space is RoomBlock)
{
    gradi = FaceB.AlphaR * (FaceB.Space.MRT -
TtmpOld[0]); //先頭材料への放射熱
qgeni =
FaceB.Space.SensibleHeatGenerationRadiationToWall;
qsoli = FaceB.Space.SolarRadiationToWall;
}
}

for (long n = 0; n < IterationNum; n++) //こ
の壁の反復
{
    for (int i = 0; i < NodeNumber; i++) //節
    点
    巡回
    {
        //I側隣接点からの熱流
        if (i == 0)
            qi = qspci + gradi + qsoli + qgeni;
        else
            qi = -qj;

        //J側隣接点からの熱流
        if (i == NodeNumber - 1)
            qj = qspcj + gradj + qsolj + qgenj;
        else
            qj = 1.0 / NodeR[i] * (TtmpOld[i + 1] -
TtmpOld[i]); //W/m2

        // 温度更新
        TtmpNew[i] = TtmpOld[i] + (qi + qj) * (Dt /
1000.0) / NodeC[i]; //Dtは[ms]なので注意
    }

    for (int i = 0; i < NodeNumber; i++)
        TtmpOld[i] = TtmpNew[i];
}

q_WL[0] = FaceA.GetHeat(); //TODO: q_r->me 相当の処
理の実装
J_WL[0] = FaceA.GetMoisture(); //TODO: q_r->me 相当
の処理の実装

q_WL[1] = (T[0] - T[1]) / (d[0] / gamma[0] + 0.5 *
d[1] / gamma[1]);
J_WL[1] = (X[0] - X[1]) / (d[0] / gamma_x[0] + 0.5 *
d[1] / gamma_x[1]);

for (int i = 2; i < n - 1; i++)
{
    q_WL[i] = (TWL[i - 1] - TWL[i]) / (0.5 * (d[i - 1]

```

```

/ gamma[i - 1] + d[i] / gamma[i]));
J_WL[i] = (XWL[i - 1] - XWL[i]) / (0.5 * (d[i - 1]
/ gamma_x[i - 1] + d[i] / gamma_x[i]));
}

q_WL[n - 1] = (TWL[n - 2] - TWL[n - 1]) / (0.5 * d[n
- 2] / gamma[n - 1] + d[n - 1] / gamma[n - 1]);
J_WL[n - 1] = (XWL[n - 2] - XWL[n - 1]) / (0.5 * d[n
- 2] / gamma_x[n - 1] + d[n - 1] / gamma_x[n - 1]);

q_WL[n] = -FaceB.GetHeat();
J_WL[n] = -FaceB.GetMoisture();

//温湿度更新
for (int i = 0; i < d.Length; i++)
{
    var AT = v * d[i] * Area;
    var AX = (0.5 * 1.2 + k) * d[i] * Area;
    var BT = (1000.0 * 700.0 + 2500.0 * v) * d[i] *
Area;
    var BX = 2500.0 * k * d[i] * Area;

    var dT = (AX * (q_WL[i] - q_WL[i + 1]) + BX *
(J_WL[i] - J_WL[i + 1])) / (AX * BT - AT * BX) *
Area;
    if (!Double.IsNaN(dT)) T[i] += dT;

    var dX = (AT * (q_WL[i] - q_WL[i + 1]) + BT *
(J_WL[i] - J_WL[i + 1])) / (AX * BT - AT * BX) *
Area;
    if (!Double.IsNaN(dX)) X[i] += dX;
}

///
///反復終了
///
for (int i = 0; i < NodeNumber; i++)
    Tnew[i] = TtmpNew[i];

base.Run(context);
}

public override void Commit(SimulationContext context)
{
    base.Commit(context);

    for (int i = 0; i < NodeNumber; i++)
        T[i] = Tnew[i];
}

public override void Init(SimulationContext context)
{
    //base.Dt = context.WallDt;
}

public void AddHeatA()
{
}

public void AddHeatB()
{
}

```

```

public void AddMoistureA()
{
}
public void AddMoistureB()
{
}
}
}
}

```

1.11 WindowBlock.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 窓
    /// </summary>
    public class WindowBlock: BaseBlock, ISurface, IGlass
    {
        #region 特性

        /// <summary>
        /// 窓面積[m2]
        /// </summary>
        public double Area { get { return Width * Height; } }

        /// <summary>
        /// 幅[m2]
        /// </summary>
        public double Width { get; private set; }

        /// <summary>
        /// 高さ[m2]
        /// </summary>
        public double Height { get; private set; }

        /// <summary>
        /// 熱貫流率
        /// </summary>
        public double UValue { get; private set; }

        /// <summary>
        ///  $\eta R$ 
        /// </summary>
        public double EtaR { get; set; }

        /// <summary>
        ///  $\eta C$ 
        /// </summary>
        public double EtaC { get; set; }

        /// <summary>
        /// ガラス厚さ[m] 熱容量算定用
        /// </summary>

```

```

public double Thickness { get; private set; }

    /// <summary>
    /// 表面A
    /// </summary>
    public WallFaceConnector FaceA { get; private set; }

    /// <summary>
    /// 表面B
    /// </summary>
    public WallFaceConnector FaceB { get; private set; }

    /// <summary>
    /// 太陽方向となす角
    /// </summary>
    public double Costheta { get; set; }

    /// <summary>
    /// 窓の計算時間間隔[ms]
    /// </summary>
    public override long Dt { get; set; }

    /// <summary>
    /// 1minにおける壁の計算回数
    /// </summary>
    private long IterationNum;

    private int NodeNumber = 2;
    private double[] T; // 節点温度
    private double[] Tnew; // 新節点温度
    private double[] X; // 節点湿度
    private double[] Xnew; // 新節点湿度
    private double NodeR; // 節点間熱抵抗
    private double[] NodeC; // 節点熱容量

    #endregion

    #region 状態値

    /// <summary>
    /// A側表面温度[°C]
    /// </summary>
    public double TempA
    {
        get
        {
            return T[0]; // 窓は必ずA側をT[0], B側をT[1]とする
        }
    }

    /// <summary>
    /// B側表面温度[°C]
    /// </summary>
    public double TempB
    {
        get
        {
            return T[1];
        }
    }

    /// <summary>
    /// A側表面絶対湿度[kg/kg']
    /// </summary>
    public double HumidityA { get; private set; }

    /// <summary>

```

```

/// B側表面絶対湿度[kg/kg']
/// </summary>
public double HumidityB { get; private set; }

#endregion

public override bool IsComposite
{
    get
    {
        return true;
    }
}

public override IConnector[] GetConnectors()
{
    return new IConnector[] { FaceA, FaceB };
}

/// <summary>
/// コンストラクタ
/// </summary>
/// <param name="area">面積[m^2]</param>
/// <param name="windowProperty">窓仕様No</param>
/// <param name="width">幅[m]</param>
/// <param name="height">高さ[m]</param>
/// <param name="faceh">面傾斜角[deg]</param>
/// <param name="facea">面方位角[deg]</param>
public WindowBlock(string name, ISpaceBlock spaceA,
ISpaceBlock spaceB, double width, double height,
double uvalue, double etar, double etac, double
thickness, double faceh, double facea) : base()
{
    this.Name = name;
    this.Width = width;
    this.Height = height;
    this.UValue = uvalue;
    this.EtaR = etar;
    this.EtaC = etac;
    this.Thickness = thickness;

    double alphacA = 4.4;
    double alphasA = 4.7;
    double alphacB = 20.3;
    double alphasB = 4.7;
    double solarabsorptionrate = 0.0;
    double longwaveemissionrate = 0.9;

    if (spaceB is RoomBlock)
    {
        alphacB = alphacA;
    }

    this.FaceA = new WallFaceConnector(name + "A",
spaceA, this, 1, 0, this.Area,
alphacA, alphasA, solarabsorptionrate,
longwaveemissionrate, faceh, facea);
    this.FaceB = new WallFaceConnector(name + "B",
spaceB, this, 2, 0, this.Area,
alphacB, alphasB, solarabsorptionrate,
longwaveemissionrate, faceh, facea);

//spaceAに登録
if (spaceA is RoomBlock)
{
    spaceA.SetFaceConnector(this.FaceA, 0);
}

```

```

//spaceBに登録
if (spaceB is RoomBlock)
{
    spaceB.SetFaceConnector(this.FaceB, 0);
}

//作業領域作成
double cr = 2268 * 1000.0; //ガラス容積比熱
[J/m3K]

NodeR = 1.0 / uvalue - 1.0 / (alphacA + alphasA) -
1.0 / (alphacB + alphasB);
NodeC = new double[NodeNumber];
for (int i = 0; i < NodeNumber; i++)
{
    NodeC[i] = cr * 0.5 * thickness;
}

//収束条件のチェック
double t = 60.0; //最大20s
double dt_tmp = 0.0; //

dt_tmp = NodeC[1] / (alphacB + alphasB + 1.0 /
NodeR); //B側が条件的に厳しい
if (dt_tmp < t) t = dt_tmp;
Dt = (long)(t * 1000.0); //[ms]単位で保持する

if (Dt == 0)
{
    System.Environment.Exit(1);
}

Dt = base.CalcDt(Dt);
IterationNum = 60000 / Dt; //20sを進めるための反
復回数

T = new double[NodeNumber];
Tnew = new double[NodeNumber];
X = new double[NodeNumber];
Xnew = new double[NodeNumber];

double initialtemp = 20.0;
double initialhumidity = 0.0;
for (int i = 0; i < NodeNumber; i++)
{
    T[i] = initialtemp;
    Tnew[i] = initialtemp;
    X[i] = initialhumidity;
    Xnew[i] = initialhumidity;
}

}

/// <summary>
/// 反復前の毎回の初期化(室への日射導入)
/// </summary>
/// <param name="context"></param>
public override void IterationInit(SimulationContext
context)
{
    ///
    /// 透過日射
    ///
    if (FaceB.Space is OutsideBlock)

```

```

{
    double costheta = FaceB.CosTheta;
    double sd = FaceB.DirectSolar;
    double ss = FaceB.SkySolar;
    double sr = FaceB.ReflectSolar;

    ISpaceBlock SpaceA = FaceA.Space;    ///スペースの取得
    double afd = AngleFactor(costheta) / 0.88;    ///入射角特性(直達) 基準化
    double afs = AngleFactor() / 0.88;    ///入射角特性(天空反射) 基準化

    ///
    ///室へ日射を与える
    ///

    SpaceA.DirectSolarRadiationTotal += sd * afd * EtaR * this.Area;
    SpaceA.DirectSolarConvectionTotal += sd * afd * EtaC * this.Area;
    SpaceA.SkySolarRadiationTotal += ss * afs * EtaR * this.Area;
    SpaceA.SkySolarConvectionTotal += ss * afs * EtaC * this.Area;
    SpaceA.ReflectSolarRadiationTotal += sr * afs * EtaR * this.Area;
    SpaceA.ReflectSolarConvectionTotal += sr * afs * EtaC * this.Area;
}

public override void Run(SimulationContext context)
{
    ///熱流・水分流
    double[] TtmpOld = new double[NodeNumber];    ///1min 進行するまでの(仮)更新後温度
    double[] TtmpNew = new double[NodeNumber];    ///1min 進行するまでの(仮)更新後温度

    for (int i = 0; i < NodeNumber; i++)
        TtmpOld[i] = T[i];

    ///I側・J側表面
    ///窓はI側=A側で固定
    double qi = 0.0;    ///i側隣接節点からの熱[W/m2]
    double qj = 0.0;    ///j側隣接節点からの熱[W/m2]
    double qrad_i = 0.0;
    double qrad_j = 0.0;
    double qgen_i = 0.0;
    double qgen_j = 0.0;

    double qspci = FaceA.GetHeat();    ///i側スペースからの対流熱[W/m2] スペースから面に入る場合を正
    double qspcj = FaceB.GetHeat();    ///j側スペースからの対流熱[W/m2] スペースから面に入る場合を正

    if (FaceA.Space is RoomBlock)
    {
        qrad_i = FaceA.AlphaR * (FaceA.Space.MRT - TtmpOld[0]);    ///i側スペースからの放射熱[W/m2]
        qgen_i = FaceA.Space.SensibleHeatGenerationRadiationToWall;
    }
}

```

```

if (FaceB.Space is RoomBlock)
{
    qrad_j = FaceB.AlphaR * (FaceB.Space.MRT - TtmpOld[NodeNumber - 1]);    ///j側スペースからの放射熱[W/m2]
    qgen_j = FaceB.Space.SensibleHeatGenerationRadiationToWall;
}

for (int i = 0; i < NodeNumber; i++)
    TtmpOld[i] = T[i];

for (long n = 0; n < IterationNum; n++)    ///この窓の反復
{
    for (int i = 0; i < NodeNumber; i++)
    {
        ///I側隣接点からの熱流
        if (i == 0)
        {
            qi = qspci + qrad_i + qgen_i;
            qj = 1.0 / NodeR * (TtmpOld[i + 1] - TtmpOld[i]);    ///W/m2
        }
        else
        {
            qi = -qj;
            qj = qspcj + qrad_j + qgen_j;
        }

        /// 温度更新
        TtmpNew[i] = TtmpOld[i] + (qi + qj) * (Dt / 1000.0) / NodeC[i];    ///Dtは[ms]なので注意
    }
}

///
///反復終了
///
for (int i = 0; i < NodeNumber; i++)
    Tnew[i] = TtmpNew[i];
}

public override void Commit(SimulationContext context)
{
    base.Commit(context);

    for (int i = 0; i < NodeNumber; i++)
        T[i] = Tnew[i];
}

public override void Init(SimulationContext context)
{
    ///base.Dt = context.WindowDt;
}

public void AddHeatA()
{
}

```



```

public void AddHeatB()
{
}

public void AddMoistureA()
{
}

public void AddMoistureB()
{
}

public double AngleFactor(double costheta)
{
    if (costheta <= 0.0)
        return 0.0;
    double a = costheta * costheta;
    return costheta * (2.392 + a * (-3.8636 + a *
        (3.7568 - 1.3952 * a)));
}

public double AngleFactor()
{
    return 0.81;
}
}

public interface IGlass
{
    double EtaR { get; set; }
    double EtaC { get; set; }
    double AngleFactor(double costheta);
}
}

```

1.12 DoorBlock.cs

```

using HLFX.Utility;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    public class DoorBlock : BaseBlock, ISurface
    {
        #region 特性

        /// <summary>
        /// 面積[m2]
        /// </summary>
        public double Area { get { return Width * Height; } }

        /// <summary>
        /// 幅[m2]
        /// </summary>
        public double Width { get; private set; }

```

```

        /// <summary>
        /// 高さ[m2]
        /// </summary>
        public double Height { get; private set; }

        /// <summary>
        /// 熱貫流率
        /// </summary>
        public double UValue { get; private set; }

        /// <summary>
        /// 材料
        /// 容積比熱のみ参照する
        /// </summary>
        public Material Material { get; }

        /// <summary>
        /// 厚さ[m] 熱容量算定用
        /// </summary>
        public double Thickness { get; private set; }

        #endregion

        /// <summary>
        /// 表面A
        /// </summary>
        public WallFaceConnector FaceA { get; private set; }

        /// <summary>
        /// 表面B
        /// </summary>
        public WallFaceConnector FaceB { get; private set; }

        /// <summary>
        /// ドアの計算時間間隔[ms]
        /// </summary>
        public override long Dt { get; set; }

        /// <summary>
        /// lminにおける壁の計算回数
        /// </summary>
        private long IterationNum;

        private int NodeNumber = 2;
        private double[] T; // 節点温度
        private double[] Tnew; // 新節点温度
        private double[] X; // 節点湿度
        private double[] Xnew; // 新節点湿度
        private double[] NodeR; // 節点間熱抵抗
        private double[] NodeC; // 節点熱容量

        public double Costheta { get; set; }
        #region 状態値

        /// <summary>
        /// A側表面温度[°C]
        /// </summary>
        public double TempA
        {
            get
            {
                return T[0]; // 窓は必ずA側をT[0], B側をT[1]とする
            }
        }

        /// <summary>

```

```

/// B側表面温度[°C]
/// </summary>
public double TempB
{
    get
    {
        return T[1];
    }
}

/// <summary>
/// A側表面絶対湿度[kg/kg']
/// </summary>
public double HumidityA { get; private set; }

/// <summary>
/// B側表面絶対湿度[kg/kg']
/// </summary>
public double HumidityB { get; private set; }

#endregion

/// <summary>
/// コンストラクタ
/// </summary>
/// <param name="area">面積[m^2]</param>
/// <param name="windowProperty">窓仕様No</param>
/// <param name="width">幅[m]</param>
/// <param name="height">高さ[m]</param>
/// <param name="faceh">面傾斜角[deg]</param>
/// <param name="facea">面方位角[deg]</param>
public DoorBlock(string name, ISpaceBlock spaceA,
ISpaceBlock spaceB, double width, double height,
double uvalue, Material material, double thickness,
double faceh, double facea) : base()
{
    this.Name = name;
    this.Width = width;
    this.Height = height;
    this.UValue = uvalue;
    this.Thickness = thickness;
    this.Material = material;

    double alphacA = 4.4;
    double alphacB = 4.4;
    double alphas = 4.7;
    double solarabsorptionrate = 0.8;
    double longwaveemissionrate = 0.9;

    if (spaceB is OutsideBlock)
    {
        alphacB = 20.3;
    }

    if (spaceA is OutsideBlock)
    {
        alphacA = 20.3;
    }

    this.FaceA = new WallFaceConnector(name + "A",
spaceA, this, 1, 0, this.Area,
alphacA, alphas, solarabsorptionrate,
longwaveemissionrate, faceh, facea);
    this.FaceB = new WallFaceConnector(name + "B",
spaceB, this, 2, 0, this.Area,
alphacB, alphas, solarabsorptionrate,
longwaveemissionrate, faceh, facea);
}

```

```

//spaceAに登録
if (spaceA is RoomBlock)
{
    spaceA.SetFaceConnector(this.FaceA, 0);
}

//spaceBに登録
if (spaceB is RoomBlock)
{
    spaceB.SetFaceConnector(this.FaceB, 0);
}

//作業領域作成
double cr = Material.容積比熱; //材料容積比熱[J/m3K]

NodeR = 1.0 / uvalue - 1.0 / (alphacA + alphas) - 1.0 / (alphacB + alphas);
NodeC = new double[NodeNumber];
for (int i = 0; i < NodeNumber; i++)
{
    NodeC[i] = cr * 0.5 * thickness;
}

//取束条件のチェック
double t = 60.0; //最大60s
double dt_tmp = 0.0; //

dt_tmp = NodeC[1] / (alphacB + alphas + 1.0 / NodeR);
//一方が外気だとすればB側。B側が条件的に厳しい
if (dt_tmp < t) t = dt_tmp;
Dt = (long)(t * 1000.0); //[ms]単位で保持する

if (Dt == 0)
{
    System.Environment.Exit(1);
}

Dt = base.CalcDt(Dt);
IterationNum = 60000 / Dt; //1minの反復回数

T = new double[NodeNumber];
Tnew = new double[NodeNumber];
X = new double[NodeNumber];
Xnew = new double[NodeNumber];

double initialtemp = 20.0;
double initialhumidity = 0.0;
for (int i = 0; i < NodeNumber; i++)
{
    T[i] = initialtemp;
    Tnew[i] = initialtemp;
    X[i] = initialhumidity;
    Xnew[i] = initialhumidity;
}

}

public override void Run(SimulationContext context)
{
    //熱流・水分流
    double[] TtmpOld = new double[NodeNumber]; //1min
    進行するまでの(仮)更新後温度
    double[] TtmpNew = new double[NodeNumber]; //1min
}

```

進行するまでの(仮)更新後温度

```
for (int i = 0; i < NodeNumber; i++)
    TtmpOld[i] = T[i];

//I側・J側表面
//窓はI側=A側で固定
double qi = 0.0; //i側隣接節点からの熱[W/m2]
double qj = 0.0; //j側隣接節点からの熱[W/m2]
double qradi = 0.0;
double qradj = 0.0;
double qgeni = 0.0;
double qgenj = 0.0;
double qsoli = 0.0;
double qsolj = 0.0;

double qspci = FaceA.GetHeat(); //i側スペースからの対流または総合熱伝達[W/m2] スペースから面に入る場合を正
double qspcj = FaceB.GetHeat(); //j側スペースからの対流または総合熱伝達[W/m2] スペースから面に入る場合を正

if (FaceA.Space is RoomBlock)
{
    qradi = FaceA.AlphaR * (FaceA.Space.MRT - TtmpOld[0]); //i側スペースからの放射熱[W/m2]
    qgeni = FaceA.Space.SensibleHeatGenerationRadiationToWall;
    qsoli = FaceA.Space.SolarRadiationToWall;
}

if (FaceB.Space is RoomBlock)
{
    qradj = FaceB.AlphaR * (FaceB.Space.MRT - TtmpOld[NodeNumber - 1]); //j側スペースからの放射熱[W/m2]
    qgenj = FaceB.Space.SensibleHeatGenerationRadiationToWall;
    qsolj = FaceB.Space.SolarRadiationToWall;
}

for (int i = 0; i < NodeNumber; i++)
    TtmpOld[i] = T[i];

for (long n = 0; n < IterationNum; n++) //このドアの反復
{
    for (int i = 0; i < NodeNumber; i++)
    {
        //I側隣接点からの熱流
        if (i == 0)
        {
            qi = qspci + qradi + qgeni + qsoli;
            qj = 1.0 / NodeR * (TtmpOld[i + 1] - TtmpOld[i]); //W/m2
        }
        else
        {
            qi = -qj;
            qj = qspcj + qradj + qgenj + qsolj;
        }
    }

    /// 温度更新
```

```
TtmpNew[i] = TtmpOld[i] + (qi + qj) * (Dt / 1000.0) / NodeC[i]; //Dtは[ms]なので注意
    }
}

///
///反復終了
///
for (int i = 0; i < NodeNumber; i++)
    Tnew[i] = TtmpNew[i];

}

public override void Commit(SimulationContext context)
{
    base.Commit(context);

    for (int i = 0; i < NodeNumber; i++)
        T[i] = Tnew[i];
}

public void AddHeatA()
{
}

public void AddHeatB()
{
}

public void AddMoistureA()
{
}

public void AddMoistureB()
{
}
}
```

1.13 WallFaceConnector.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 壁面・窓面
    /// </summary>
    public class WallFaceConnector: ILayerBlock, IBlock
    {
        public bool IsComposite { get { return false; } }

        public long Dt { get; set; }
    }
}
```

```

public IConnector[] GetConnectors()
{
    return null;
}

#region 接続
/// <summary>
/// 接続するオブジェクト(居室または外部環境)
/// </summary>
public ISpaceBlock Space { get; private set; }

/// <summary>
/// 接続するオブジェクト(壁/窓/ドアのいずれか)
/// </summary>
public ISurface Surface { get; private set; }

/// <summary>
/// オブジェクト(壁または窓)のどちら側の表面に接続する
/// か(1:A側, 2:B側) 材料のI側J側ではないので注意
/// </summary>
public int SurfaceDirection { get; private set; }

#endregion

#region 特性
/// <summary>
/// 壁・窓表面積[m2]
/// </summary>
public double Area { get; private set; }

/// <summary>
/// 対流熱伝達率[W/m2K]
/// </summary>
public double AlphaC { get; private set; }

/// <summary>
/// 放射熱伝達率[W/m2K]
/// </summary>
public double AlphaR { get; private set; }

/// <summary>
/// 総合熱伝達率[W/m2K]
/// </summary>
public double AlphaT
{
    get { return AlphaC +AlphaR; }
}

public double FaceSinH { get; set; }
public double FaceSinA { get; set; }
public double FaceCosH { get; set; }
public double FaceCosA { get; set; }
private double FaceWz;
private double FaceWw;
private double FaceWs;

///public double CosTheta { get; set; }

/// <summary>
/// 日射吸収率[-]
/// </summary>
public double SolarAbsorptionRate { get; private set; }

/// <summary>
/// 長波長放射率[-]

```

```

/// </summary>
public double LongWaveEmissionRate { get; private set; }

/// <summary>
/// 水分伝達率[kg/m2s(kg/kg' )]
/// </summary>
public double MoistureTransferRate { get; private set; }

#endregion

#region 状態値
/// <summary>
/// スペース側の空気温度[°C]
/// 熱伝達計算時に呼び出される
/// 外気(総合熱伝達)の場合は表面にcosθ, 入射日射量(直
/// 達・天空・反射)もセットする
/// </summary>
public double SpaceTemp
{
    get
    {
        if (Space is RoomBlock)
            return Space.Temp;
        else
        {
            double costheta = Space.GetCosTheta(FaceWz,
            FaceWw, FaceWs);
            Surface.Costheta = costheta;

            if (Surface is WallBlock || Surface is DoorBlock)
                return Space.GetSAT(SolarAbsorptionRate,
                LongWaveEmissionRate, AlphaT, costheta,
                FaceCosH);
            else if (Surface is WindowBlock)
                return Space.GetSAT(LongWaveEmissionRate, AlphaT,
                costheta, FaceCosH);
            else
                return Space.Temp;
        }
    }
}

/// <summary>
/// 絶対湿度 [kg/kg']
/// </summary>
public double SpaceHumidity
{
    get
    {
        return Space.Humidity;
    }
}

/// <summary>
/// 入射する直達日射[W/m2]
/// </summary>
public double DirectSolar { get; private set; }

/// <summary>
/// 入射する天空日射[W/m2]
/// </summary>
public double SkySolar { get; private set; }

/// <summary>

```

```

/// 入射する反射日射[W/m2]
/// </summary>
public double ReflectSolar { get; private set; }

/// <summary>
/// 放出する夜間放射[W/m2]
/// </summary>
public double LongWave { get; private set; }

/// <summary>
/// 太陽方向となす角のcos
/// </summary>
public double CosTheta { get; private set; }

/// <summary>
/// 表面温度[°C]
/// </summary>
public double SurfaceTemp
{
    get
    {
        if (SurfaceDirection == 1) return Surface.TempA;
        else return Surface.TempB;
    }
}

/// <summary>
/// 絶対湿度 [kg/kg']
/// </summary>
public double SurfaceHumidity
{
    get
    {
        if (SurfaceDirection == 1) return
            Surface.HumidityA;
        else return Surface.HumidityB;
    }
}

public string Name { get; set; }

/// <summary>
/// 対流熱流量 単位注意[W/m2]
/// </summary>
private double Qc;

private double dJ_tick;

#endregion

/// <summary>
/// 壁計算オブジェクト
/// </summary>
private int layerIndex;

/// <summary>
/// コンストラクタ
/// </summary>
/// <param name="space">居室または外部</param>
/// <param name="surface">壁または窓</param>
/// <param name="surfacedirection">壁または窓の接続方
/// 向 (A側=1, B側=2)</param>

```

```

/// <param name="area">面積[m^2]</param>
/// <param name="alphac">対流熱伝達率[W/m2K]</param>
/// <param name="alphan">放射熱伝達率[W/m2K]</param>
/// <param name="solarabsorptionrate">日射吸収率
[-]</param>
/// <param name="longwaveemissionrate">長波放射率
[-]</param>
/// <param name="layerIndex">壁計算オブジェクトにおけ
る層インデックス</param>
/// <param name="faceh">面傾斜角[deg]</param>
/// <param name="facea">面方位角[deg]</param>
public WallFaceConnector(string name, ISpaceBlock
space, ISurface surface, int surfacedirection, int
layerIndex,
double area, double alphac, double alphan, double
solarabsorptionrate, double longwaveemissionrate,
double faceh, double facea)
{
    this.Name = name;
    this.Area = area;
    this.AlphaC = alphac;
    this.AlphaR = alphan;
    this.SolarAbsorptionRate = solarabsorptionrate;
    this.LongWaveEmissionRate = longwaveemissionrate;
    this.MoistureTransferRate = alphac * 0.01;

    this.Space = space;
    this.Surface = surface;
    this.SurfaceDirection = surfacedirection;
    this.LayerIndex = layerIndex;

    this.FaceSinH = Math.Sin(faceh / 180.0 * Math.PI);
    this.FaceCosH = Math.Cos(faceh / 180.0 * Math.PI);
    this.FaceSinA = Math.Sin(facea / 180.0 * Math.PI);
    this.FaceCosA = Math.Cos(facea / 180.0 * Math.PI);
    this.FaceWz = this.FaceCosH;
    this.FaceWw = this.FaceSinH * this.FaceSinA;
    this.FaceWs = this.FaceSinH * this.FaceCosA;
}

public void Init(SimulationContext context)
{
}

/// <summary>
/// 初期化 (反復前に毎回)
/// </summary>
/// <param name="context"></param>
public void IterationInit(SimulationContext context)
{
    //
    //太陽とのなす角,表面の受ける日射量等を取得する
    //

    if (Space is OutsideBlock)
    {
        double sd = 0.0;
        double ss = 0.0;
        double sr = 0.0;
        double lw = 0.0;

        CosTheta = Space.GetCosTheta(FaceWz, FaceWw,
FaceWs);
        Space.GetSolar(CosTheta, FaceCosH, out sd, out ss,
out sr, out lw);
    }
}

```

```

    DirectSolar = sd;
    SkySolar = ss;
    ReflectSolar = sr;
    LongWave = lw;
}
}

public void Run(SimulationContext context)
{
}

public void Commit(SimulationContext context)
{
}

/// <summary>
/// 対流熱流量[W/m2] スペースから面に向かう場合を正
/// </summary>
public virtual void AddHeat(double qc) ///
{
    Qc = qc;
}

/// <summary>
/// 対流熱流量[W/m2] スペースから面に向かう場合を正
/// </summary>
public virtual double GetHeat()
{
    return Qc;
}

/// <summary>
/// 水分移動
/// </summary>
/// <param name="dJ">水分伝達量 (単位[kg]) </param>
public virtual void AddMoisture(double dJ, IConnector
src, IConnector kernel)
{
    dJ_tick += dJ;
    System.Diagnostics.Debug.Assert(dJ != -0.0002);

    System.Diagnostics.Debug.WriteLine("AddHeat: [{0}]-
>[{1}] {2}[kg] by {3}", src.Name, this.Name, dJ,
kernel);
}

public virtual void AddMoisture(double dJ)
{
    dJ_tick += dJ;
}

public virtual double GetMoisture()
{
    return dJ_tick;
}
}
}

```

1.14 OutsideBlock.cs

```
using HLFX.Utility;
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 外部環境
    /// </summary>
    public class OutsideBlock: BaseBlock, ISpaceBlock
    {
        /// <summary>
        /// 外気温度
        /// </summary>
        public double Temp { get; set; }

        /// <summary>
        /// 絶対湿度[kg/kg']
        /// </summary>
        public double Humidity { get; set; }

        /// <summary>
        /// 法線面直達日射量[W/m2]
        /// </summary>
        public double DSolar { get; set; }

        /// <summary>
        /// 水平面天空日射量[W/m2]
        /// </summary>
        public double SSolar { get; set; }

        /// <summary>
        /// 夜間放射量[W/m2]
        /// </summary>
        public double Lradiation { get; set; }

        private double SinH;
        private double CosH;
        private double SinA;
        private double CosA;

        /// <summary>
        /// 反射日射量(下向き水平面仮定) [W/m2]
        /// </summary>
        private double RSolar;

        /// <summary>
        /// ダミー
        /// </summary>
        public double MRT { get; set; }
        public double DirectSolarRadiationTotal { get; set; }
        public double DirectSolarConvectionTotal { get; set; }

        public double SkySolarRadiationTotal { get; set; }
        public double SkySolarConvectionTotal { get; set; }
        public double ReflectSolarRadiationTotal { get; set; }
        public double ReflectSolarConvectionTotal { get;
set; }

        public double SensibleHeatGenerationRadiationToWall
{ get { return 0.0; } }

        public double SolarRadiationToFloor { get { return
0.0; } }
    }
}

```

```

public double SolarRadiationToWall { get { return
0.0; } }

/// <summary>
/// ファイル名
/// </summary>
public string FileName { get; set; }

/// <summary>
/// 気象データ
/// </summary>
private IList<SMASHWeatherData> dataSet;

public OutsideBlock()
{
    Name = "外部環境";
}

private void Update(DateTime dt)
{
    //補間値の取得
    var data =
SMASHWeatherLoader.GetLinearInterpolated(dt.Month,
dt.Day, dt.Hour, dt.Minute, dataSet);

    //設定
    Temp = data.Temperature;
    Humidity = data.Humidity;
    DSolar = data.Ids;
    SSolar = data.Iis;
    Lradiation = data.Ins;
    SinH = data.SinH;
    SinA = data.SinA;
    CosH = data.CosH;
    CosA = data.CosA;

    RSolar = 0.1 * (DSolar * SinH + SSolar);
}

/// <summary>
/// 面の角度を受け取り, 太陽方向となす角のcosを返す
/// </summary>
/// <param name="facewz">面のcos(傾斜角)</param>
/// <param name="faceww">面のsin(傾斜角) × sin(方位
角)</param>
/// <param name="faceww">面のsin(傾斜角) × cos(方位
角)</param>
public double GetCosTheta(double facewz, double
faceww, double facews)
{
    return this.SinH * facewz + this.CosH * this.SinA *
faceww + this.CosH * this.CosA * facews;
}

/// <summary>
/// 壁の場合のSAT温度を返す
/// </summary>
/// <param name="solarabsorptionrate">表面の日射吸収率
</param>
/// <param name="longwaveemissionrate">表面の長波放射
率</param>
/// <param name="alphan">表面の $\alpha t$ </param>
/// <param name="costheta">面と太陽方向の $\cos \theta$ </param>
/// <param name="facecosh">面の $\cos H$ </param>
public double GetSAT(double solarabsorptionrate,
double longwaveemissionrate, double alphan, double
costheta, double facecosh)

```

```

{
    double a = 0.0;
    double sf = 0.5 * (1.0 + facecosh);    ///天空形態係
    数

    if (costheta > 0.0)
        a += solarabsorptionrate * DSolar * costheta;

    a += solarabsorptionrate * (sf * SSolar + (1.0 - sf)
* RSolar);

    a -= longwaveemissionrate * sf * Lradiation;

    if (alphan > 0.0)
        a = a / alphan + Temp;
    else
        a = Temp;

    return a;
}

/// <summary>
/// 窓の場合のSAT温度を返す
/// </summary>
/// <param name="longwaveemissionrate">表面の長波放射
率</param>
/// <param name="alphan">表面の $\alpha t$ </param>
/// <param name="costheta">面と太陽方向の $\cos \theta$ </param>
/// <param name="facecosh">面の $\cos H$ </param>
public double GetSAT(double longwaveemissionrate,
double alphan, double costheta, double facecosh)
{
    double a = 0.0;
    double sf = 0.5 * (1.0 + facecosh);    ///天空形態係
    数

    a -= longwaveemissionrate * sf * Lradiation;

    if (alphan > 0.0)
        a = a / alphan + Temp;
    else
        a = Temp;

    return a;
}

/// <summary>
/// 面の角度を受け取り, 当該面に入射する日射量, 夜間放射
量を返す
/// </summary>
public void GetSolar(double costheta, double facecosh,
out double directsolar, out double skysolar, out
double reflectsolar, out double longwave)
{
    double sf = 0.5 * (1.0 + facecosh);    ///面からみた天
    空形態係数

    if (costheta > 0.0)
        directsolar = DSolar * costheta;
    else
        directsolar = 0.0;

    skysolar = sf * SSolar;
    reflectsolar = (1.0 - sf) * RSolar;
    longwave = sf * Lradiation;
}

```

```

public override void Init(SimulationContext context)
{
    //気象データ読み込み
    dataSet = SMASHWeatherLoader.ReadAllData(fileName);

    Update(context.Current);

    base.Init(context);
}

public override void IterationInit(SimulationContext
context)
{
    Update(context.Current);
}

public override void Commit(SimulationContext context)
{
    ///Update(context.Current);

    base.Commit(context);
}

public void SetFaceConnector(WallFaceConnector
connectingface, int facetype)
{
    ///何もしない
}

public void
SetVentilation(VentilationHeatTransferBlock
ventilation)
{
    ///何もしない
}

public void AddVent(double qv, double jv)
{
    ///何もしない
}
}
}

```

1.15 FixedConditionRoomBlock.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    public class FixedConditionRoomBlock : RoomBlock,
    ISpaceBlock
    {
        public struct ReferenceSpace
        {
            public double Ratio; //重み
            public ISpaceBlock Space;
        }
    }
}

```

```

public ReferenceSpace(ISpaceBlock space, double
ratio)
{
    Space = space;
    Ratio = ratio;
}

/// <summary>
/// 参照するスペースと比率
/// </summary>
private List<ReferenceSpace> ListReferenceRoom = new
List<ReferenceSpace> ();

public double RatioSum { get; private set; }

public void SetReferencespace(ISpaceBlock
referencespace, double ratio)
{
    ReferenceSpace space = new
ReferenceSpace(referencespace, ratio);
    ListReferenceRoom.Add(space);
    RatioSum += ratio;
}

/// <summary>
/// コンストラクタ
/// </summary>
public FixedConditionRoomBlock(string name,
IList<ISpaceBlock> referencespaces, IList<Double>
ratios) : base()
{
    base.Name = name;

    for (int i = 0; i < referencespaces.Count; i++)
    {
        SetReferencespace(referencespaces[i], ratios[i]);
    }
}

public FixedConditionRoomBlock() : base()
{
    RatioSum = 0.0;
}

public override void Init(SimulationContext context)
{
    Temp = 20.0;
    Humidity = 0.010;

    Qvent = 0.0;
    Jvent = 0.0;
    Qac = 0.0;
    Jac = 0.0;
    DirectSolarRadiationTotal = 0.0;
    DirectSolarConvectionTotal = 0.0;
    SkySolarRadiationTotal = 0.0;
    SkySolarConvectionTotal = 0.0;
    ReflectSolarRadiationTotal = 0.0;
    ReflectSolarConvectionTotal = 0.0;
    SensibleHeatGenerationTotalConvection = 0.0;
    SensibleHeatGenerationTotalRadiation = 0.0;
    VaporGenerationTotal = 0.0;
}

```



```

public override void IterationInit(SimulationContext
context)
{
    if (RatioSum > 0.0)
    {
        double tempsum = 0.0;
        double humsum = 0.0;
        for (int i = 0; i < ListReferenceRoom.Count; i++)
        {
            tempsum += ListReferenceRoom[i].Space.Temp *
ListReferenceRoom[i].Ratio;
            humsum += ListReferenceRoom[i].Space.Humidity *
ListReferenceRoom[i].Ratio;
        }

        Temp = tempsum / RatioSum;
        Humidity = humsum / RatioSum;
    }
}

public override void Run(SimulationContext context)
{
}

public override void Commit(SimulationContext context)
{
}
}
}

```

1.16 PersonBlock.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 室ごとの在室人数
    /// 1人につき1インスタンスではなく、人のいる室1つにつき1
    /// インスタンスとする
    /// 複数室には適用しない
    /// </summary>
    public class PersonBlock : BaseBlock
    {
        private const double EvpHeat = 2510.0;

        public RoomBlock Room { get; set; } = null;

        public bool InRoom { get; private set; }

        public string name { get; private set; }

        /// <summary>
        /// 1人あたり全熱[W/人]
        /// </summary>

```

```

public double MaxTotalHeat { get; set; }

/// <summary>
/// 作用温度24°Cのときの顕熱[W/人]
/// </summary>
public double StdSensibleHeat { get; set; }

/// <summary>
/// 顕熱勾配(24°Cとの差に乗じる係数)
/// </summary>
public double SensibleHeatSlope { get; set; }

/// <summary>
/// 対流成分比率[-]
/// </summary>
public double ConvectionRate { get; set; }

/// <summary>
/// スケジュール
/// </summary>
public ScheduleSeries[, ] PersonSchedule;

/// <summary>
/// コンストラクタ
/// </summary>
public PersonBlock(double maxtotalheat, double
stdsensibleheat, double sensibleheatslope, double
convectionrate,
ScheduleSeries[,] schedule, RoomBlock room
)
{
    MaxTotalHeat = maxtotalheat;
    StdSensibleHeat = stdsensibleheat;
    SensibleHeatSlope = sensibleheatslope;
    ConvectionRate = convectionrate;

    PersonSchedule = new ScheduleSeries[3, 2];

    for (int term = 0; term < 3; term++)
    {
        for (int wd = 0; wd < 2; wd++)
        {
            PersonSchedule[term, wd] = schedule[term, wd];
        }
    }

    Room = room;
    room.SetPerson(this);
    this.Name = room.Name;
}

/// <summary>
/// ある時点(期,曜日,時刻,分)を受け取り、顕熱発熱量・潜
/// 熱発熱量を返す
/// <param name="time">時刻・分を4桁整数化したもの
</param>
/// <param name="qgenc">発熱量(対流成分) [W]</param>
/// <param name="qgenr">発熱量(放射成分) [W]</param>
/// <param name="jgen">水分量[kg/s]</param>
/// </summary>
public void GetHeat(SimulationContext context, out
double qgenc, out double qgenr, out double jgen)
{
    double scvalue = PersonSchedule[context.CurrentTerm,
context.CurrentWeek].GetScheduleValue(context.Current
Time);

```

```

double ot = 0.5 * (Room.Temp + Room.MRT);
double sensible = StdSensibleHeat - SensibleHeatSlope
* (ot - 24.0);

if (sensible > MaxTotalHeat) sensible = MaxTotalHeat;
if (sensible < 0.0) sensible = 0.0;

qgenc = sensible * ConvectionRate * scvalue;
qgenr = sensible * (1.0 - ConvectionRate) * scvalue;
jgen = (MaxTotalHeat - sensible) * scvalue / EvpHeat
/ 1000.0;    ///

```

```

}

else if (windowopen_before == 1 && aconoff_before
== 0)    ///

```

1.17 AirConditionerBlock.cs

```

using static HLFX.Utility.Util;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    public class AirConditionerBlock : BaseBlock
    {
        /// <summary>
        /// 空気比熱[J/kgK] 当面、定数とする
        /// </summary>
        private const double c_air = 1005;

        /// <summary>

```

```

/// 空気密度[kg/m3] 当面、定数とする
/// </summary>
private const double rho_air = 1.2;

/// <summary>
/// 蒸発潜熱[J/kg] 当面、定数とする
/// </summary>
private const double rL = 2510 * 1000;

/// <summary>
/// バイパスファクター[-]
/// </summary>
public double BypassFactor { get; private set; }

/// <summary>
/// 設定吹き出し風量[m^3/s]
/// </summary>
public double AirVolume { get; private set; }

/// <summary>
/// エアコンの冷房最大能力[kW]
/// </summary>
public double MaxCoolingCapacity { get; private set; }

/// <summary>
/// エアコンの動作タイプ
/// 0:スケジュールどおり, 1:スケジュールどおり&温度が
/// 設定値未満ならOFF, 2:人体判定併用
/// </summary>
public int WorkType { get; private set; }

#region 状態

/// <summary>
/// 時刻Tの運転状態
/// </summary>
public bool currentOperation { get; set; }

/// <summary>
/// 時刻T+1の運転状態
/// </summary>
private bool nextOperation;

/// <summary>
/// 運転状態
/// </summary>
public bool Operation
{
    get
    {
        return currentOperation;
    }
    set
    {
        nextOperation = value;
    }
}

/// <summary>
/// 顕熱処理量[W] 冷房を正
/// </summary>
public double SensibleHeat { get; private set; }

/// <summary>
/// 潜熱処理量[kg/s] 冷房(除湿)を正 [W]に戻すにはrLを
/// 乗じる
/// </summary>
public double LatentHeat { get; private set; }

/// <summary>
/// 吹き出し温度
/// </summary>
public double OutTemp { get; private set; }

/// <summary>
/// 吹き出し湿度[kg/kg']
/// </summary>
public double OutHumidity { get; private set; }

/// <summary>
/// スケジュール
/// </summary>
public ScheduleSeries[,] ACSchedule;

#endregion

#region 接続

/// <summary>
/// 居室
/// </summary>
public RoomBlock Room { get; private set; }

/// <summary>
/// 外部環境
/// </summary>
public OutsideBlock Outside { get; private set; }

#endregion

#region 計算結果

/// <summary>
/// 消費電力[kWh]
/// </summary>
public double E_AC { get; private set; }

/// <summary>
/// (計算過程)エアコンの理論効率[-]
/// </summary>
public double Efficiency { get; private set; }

/// <summary>
/// (計算過程)圧縮効率理論効率[-]
/// </summary>
public double EtaComp { get; private set; }

/// <summary>
/// (計算過程)COP[-]
/// </summary>
public double COP
{
    get
    {
        return Efficiency * EtaComp;
    }
}

#endregion

/// <summary>
/// コンストラクタ

```

```

/// </summary>
/// <param name="name">名称</param>
/// <param name="room">居室</param>
/// <param name="outside">外部環境</param>
public AirConditionerBlock(string name, RoomBlock
room, OutsideBlock outside,
double maxheatingcapacity, double maxcoolingcapacity,
double airvolume, double bypassfactor,
ScheduleSeries[,] schedule)
{
base.Name = name;
this.Room = room;
this.Outside = outside;
this.MaxCoolingCapacity = maxcoolingcapacity;
this.AirVolume = airvolume;
this.BypassFactor = bypassfactor;

ACSchedule = new ScheduleSeries[3, 2];

for (int term = 0; term < 3; term++)
{
for (int wd = 0; wd < 2; wd++)
{
ACSchedule[term, wd] = schedule[term, wd];
}
}

room.SetAirConditioner(this);
}

/// <summary>
/// 運転状態をセットする
/// 引数intのみ:人体判定結果を受け取る
/// </summary>
public void SetOperation(int aconoff)
{
if (aconoff == 1)
currentOperation = true;
else
currentOperation = false;
}

/// <summary>
/// 運転状態をセットする
/// 引数contextのみ:スケジュールのみに依存
/// </summary>
public void SetOperation(SimulationContext context)
{
double scvalue = ACSchedule[context.CurrentTerm,
context.CurrentWeek].GetScheduleValue(context.Current
Time);

if (scvalue > 0.0)
currentOperation = true;
else
currentOperation = false;
}

/// <summary>
/// 運転状態をセットする
/// 引数contextと室温
/// </summary>
public void SetOperation(SimulationContext context,
double roomtemp)
{
double scvalue = ACSchedule[context.CurrentTerm,
context.CurrentWeek].GetScheduleValue(context.Current

```

```

Time);
double reftemp = 0.0; //ONOFF判定の参照温度
(WorkType=0ならこのまま使う)

if (WorkType == 1) reftemp = roomtemp;

if (scvalue >= reftemp)
currentOperation = true;
else
currentOperation = false;
}

public override void Run(SimulationContext context)
{
///
///エアコンONの冷房期のみ動作
///
if (currentOperation == true && context.CurrentTerm
== 1)
{
#region 1. 前提条件

double roomtemp = Room.Temp;
///室空気温度
double roomabstemp =
ToAbsoluteTemperature(roomtemp); //室空気絶対温
度
double roomah = Room.Humidity;
///室空気絶対湿度[kg/kg']
double roomrh = CalcHumidity(roomtemp, roomah, 2);
///室空気相対湿度[-]

#endregion

double Ls = 0.0; //顕熱処理量[W] テーブルから補
間
double LI = 0.0; //潜熱処理量[W] テーブルから補
間

GetProcessLoad(roomtemp, roomrh, out Ls, out LI);

SensibleHeat = Ls;
LatentHeat = LI / rL; ///[W]を[kg/s]に換算

#endregion

#region 3.1 吹出し温湿度の算出

OutTemp = roomtemp - Ls / (c_air * rho_air *
AirVolume);
OutHumidity = roomah - LI / (rL * rho_air *
AirVolume);

#endregion

#region 3.2 蒸発温度の算出

///エアコンの蒸発温度 T_evap[°C]
double evptemp = (OutTemp - roomtemp *
BypassFactor) / (1.0 - BypassFactor); ///[°C]
double absevpptemp = ToAbsoluteTemperature(evptemp);
///[K]

#endregion

```

```

#region 3.3 凝縮温度の算出

double cndtemp = Outside.Temp;

#endregion

#region 3.4 理論効率の算出
// ***h_cmp_in***
double temp_cmp_in = evptemp; //圧縮
機入口温度 T_cmp_in[°C]
double pres_evap = f_P_sgase(evptemp); //蒸発
圧力 P_evap [MPa]
double pres_cmp_in = pres_evap; //圧縮
機入口圧力 P_cmp_in [MPa]
double h_cmp_in = f_H_gas_comp_in(pres_cmp_in,
temp_cmp_in); //圧縮機入口エンタルピー
h_cmp_in[kJ/kg]

// ***h_cmp_out***
double S_cmp_in = f_S_gas(pres_cmp_in, h_cmp_in);
//圧縮機入口エンタルピー S_cmp_in [kJ/kgK]
double S_cmp_out = S_cmp_in;
//圧縮機出口エンタルピー S_cmp_out [kJ/kgK]
double pres_cnd = f_P_sgase(cndtemp);
//凝縮圧力 P_cnd [MPa]
double P_cmp_out = pres_cnd;
//圧縮機出口圧力 P_cmp_out [MPa]
double h_cmp_out = f_H_gas_comp_out(P_cmp_out,
S_cmp_out); //圧縮機出口エンタルピー h_cmp_out
[kJ/kg]

// ***h_evap_in***
double temp_cnd_out = cndtemp;
//凝縮器出口温度 T_cnd_out [°C]
double h_cnd_out = f_H_liq(pres_cnd, temp_cnd_out);
//凝縮器出口エンタルピー h_cnd_out [kJ/kg]
double h_evap_in = h_cnd_out; //蒸
発器入口エンタルピー h_evap_in [kJ/kg]

// ***h_evap_out***
double h_evap_out = h_cmp_in; //蒸
発器出口エンタルピー h_evap_out [kJ/kg]

//エアコンの理論効率 e_AC
double e_AC = 0.0;

if (h_cmp_out - h_cmp_in > 0.0)
    e_AC = Math.Min((h_evap_out - h_evap_in) /
(h_cmp_out - h_cmp_in), 10.0);
else
    e_AC = 10.0;

this.Efficiency = e_AC;

#endregion

#region 3.5 消費電力の算出

//***圧縮機の消費電力E_cmp [kW]***
double qc = (Ls + LI) / 1000.0;
//処理能力[kW]
double eta = -0.1255 * qc * qc + 0.6902 * qc +
0.0104; //圧縮機の圧縮効率[-]

if (eta <= 0.1)
    eta = 0.1;

```

```

this.EtaComp = eta;

double E_cmp = qc / (eta * e_AC);
//圧縮機の消費電力E_cmp[kW]

//***補機の消費電力E_aux[kW]***
double E_aux = 0.0451;

//エアコンの消費電力 E_AC [kWh]
E_AC = (E_cmp + E_aux) * context.Dt / 3600.0;

#endregion

//熱・水分の移動
Room.AddAC(SensibleHeat, LatentHeat);
}
else
    return;
}

private const double _dT_RM = 3.0;
private const double _dh_RM = 0.10;
private static readonly double[] _T_RM = new double[]
{ 24, 27, 30, 33, 9999 };
private static readonly double[] _h_RM = new double[]
{ 0.5, 0.6, 0.7, 0.8, 9999 };

/// <summary>
/// テーブル
/// </summary>
private static readonly double[,] _Ls = new double[,]
{
    {300, 1011, 1943, 1864, 9999},
    {300, 848, 1658, 1569, 9999},
    {300, 688, 1399, 1304, 9999},
    {300, 524, 1162, 1062, 9999},
    {9999, 9999, 9999, 9999, 9999},
};

/// <summary>
/// テーブル
/// </summary>
private static readonly double[,] _LI = new double[,]
{
    { 0, 0, 1357, 1436, 9999},
    { 0, 0, 1642, 1731, 9999},
    { 0, 154, 1901, 1996, 9999},
    { 0, 349, 2138, 2238, 9999},
    {9999, 9999, 9999, 9999, 9999},
};

/// <summary>
/// 顕熱・潜熱処理量の算出
/// </summary>
/// <param name="T_RM">吸込み空気温度[°C]</param>
/// <param name="h_RM">吸込み空気相対湿度[-]</param>
/// <param name="Ls">顕熱処理量[W]</param>
/// <param name="LI">潜熱処理量[W]</param>
public void GetProcessLoad(double roomtemp, double
roomrh, out double Ls, out double LI)
{
    double T_RM = Math.Max(_T_RM[0], Math.Min(roomtemp,
_T_RM[3])); //roomtempが_T_RMの範囲内ならroomtemp,
範囲外なら適用する_T_RM
    double h_RM = Math.Max(_h_RM[0], Math.Min(roomrh,
_h_RM[3])); //roomrhが_h_RMの範囲内ならroomrh, 範
囲外なら適用する_h_RM

```

```

int j = Array.FindIndex(_T_RM, x => T_RM < x) - 1;
int i = Array.FindIndex(_h_RM, x => h_RM < x) - 1;

double Ls_low = (_Ls[i, j + 1] - _Ls[i, j]) * (T_RM -
_T_RM[j]) / _dT_RM + _Ls[i, j];
double Ls_high = (_Ls[i + 1, j + 1] - _Ls[i + 1, j])
* (T_RM - _T_RM[j]) / _dT_RM + _Ls[i + 1, j];
Ls = (Ls_high - Ls_low) * (h_RM - _h_RM[i]) / _dh_RM
+ Ls_low;

double LI_low = (_LI[i, j + 1] - _LI[i, j]) * (T_RM -
_T_RM[j]) / _dT_RM + _LI[i, j];
double LI_high = (_LI[i + 1, j + 1] - _LI[i + 1, j])
* (T_RM - _T_RM[j]) / _dT_RM + _LI[i + 1, j];
LI = (LI_high - LI_low) * (h_RM - _h_RM[i]) / _dh_RM
+ LI_low;

//
//調整が必要(もともと相対湿度50%未満の場合)
//
//顕熱のみ処理した温度で、相対湿度が90%未満の場合は潜
熱処理しない
//95%超の場合は90%まで下げる潜熱処理とする
//

if (roomrh < 0.5)
{
double temp = roomtemp - Ls / (c_air * rho_air *
AirVolume);
double abshumidity =
Utility.Util.CalcHumidity(temp, roomrh, 1);

double rh = Utility.Util.CalcHumidity(temp,
abshumidity, 2); //顕熱だけ処理した場合の相対湿
度

if (rh < 0.9)
LI = 0.0; //相対湿度90%未満の場合は潜熱処
理せず吹き出す
else
{
double newabshumidity =
Utility.Util.CalcHumidity(temp, 0.9, 1); //相対湿
度90%となる絶対湿度
LI = rL * rho_air * AirVolume * (abshumidity -
newabshumidity);
}
}
}

/// <summary>
/// 付録A.7.1 式(13)
/// </summary>
/// <param name="T">冷媒の温度[°C]</param>
/// <returns>飽和蒸気の圧力[MPa]</returns>
private double f_P_sgas(double T)
{
return
2.75857926950901 * Math.Pow(10, -17) * Math.Pow(T,
8)
+ 1.49382057911753 * Math.Pow(10, -15) *
Math.Pow(T, 7)
+ 6.52001687267015 * Math.Pow(10, -14) *
Math.Pow(T, 6)
+ 9.14153034999975 * Math.Pow(10, -12) *
Math.Pow(T, 5)

```

```

+ 3.18314616500361 * Math.Pow(10, -9) * Math.Pow(T,
4)
+ 1.60703566663019 * Math.Pow(10, -6) * Math.Pow(T,
3)
+ 3.06278984019513 * Math.Pow(10, -4) * Math.Pow(T,
2)
+ 2.54461992992037 * Math.Pow(10, -2) * Math.Pow(T,
1)
+ 7.98086455154775 * Math.Pow(10, -1)
;
}

/// <summary>
/// 付録A.8.2 式(14)
/// </summary>
/// <param name="P">過熱蒸気の圧力[MPa]</param>
/// <param name="T">過熱蒸気の温度[°C]</param>
/// <returns>過熱蒸気の比エンタルピー[kJ/kg]</returns>
private double f_H_gas_comp_in(double P, double T)
{
var K = T + 273.15;
var K2 = K * K;
var K3 = K2 * K;

var P2 = P * P;
var P3 = P2 * P;
var P4 = P2 * P2;

return
-1.00110355 * Math.Pow(10, -1) * P3
- 1.184450639 * Math.Pow(10, 1) * P2
- 2.052740252 * Math.Pow(10, 2) * P
+ 3.20391 * Math.Pow(10, -6) * K3
- 2.24685 * Math.Pow(10, -3) * K2
+ 1.279436909 * K
+ 3.1271238 * Math.Pow(10, -2) * P2 * K
- 1.415359 * Math.Pow(10, -3) * P * K2
+ 1.05553912 * P * K
+ 1.949505039 * Math.Pow(10, 2)
;
}

/// <summary>
/// 付録A.8.2 式(15)
/// </summary>
/// <param name="P">過熱蒸気の圧力[MPa]</param>
/// <param name="T">過熱蒸気の比エントロピー[kJ/(kg・
K)]</param>
/// <returns>過熱蒸気の比エンタルピー[kJ/kg]</returns>
private double f_H_gas_comp_out(double P, double S)
{
var P2 = P * P;
var P3 = P2 * P;
var P4 = P2 * P2;

var S2 = S * S;
var S3 = S2 * S;
var S4 = S2 * S2;

return
-1.869892835947070 * 0.1 * P4
+ 8.223224182177200 * 0.1 * P3
+ 4.124595239531860 * P2
- 8.346302788803210 * 10 * P
- 1.016388214044490 * 100 * S4
+ 8.652428629143880 * 100 * S3

```

```

- 2. 574830800631310 * 1000 * S2
+ 3. 462049327009730 * 1000 * S
+ 9. 209837906396910 * 0.1 * P3 * S
- 5. 163305566700450 * 0.1 * P2 * S2
+ 4. 076727767130210 * P * S3
- 8. 967168786520070 * P2 * S
- 2. 062021416757910 * 10 * P * S2
+ 9. 510257675728610 * 10 * P * S
- 1. 476914346214130 * 1000
;
}

/// <summary>
/// 付録A. 8. 2 式(16)
/// </summary>
/// <param name="P">過熱蒸気の圧力[MPa]</param>
/// <param name="h">過熱蒸気の比エンタルピー
[kJ/kg]</param>
/// <returns>過熱蒸気のエントロピー
[kJ/(kg*K)]</returns>
private double f_S_gas(double P, double h)
{
    var P2 = P * P;
    var P3 = P2 * P;
    var P4 = P2 * P2;

    var h2 = h * h;
    var h3 = h2 * h;
    var h4 = h2 * h2;

    return
        5. 823109493752840 * Math.Pow(10, -2) * P4
        - 3. 309666523931270 * Math.Pow(10, -1) * P3
        + 7. 700179914440890 * Math.Pow(10, -1) * P2
        - 1. 311726004718660 * P
        + 1. 521486605815750 * Math.Pow(10, -9) * h4
        - 2. 703698863404160 * Math.Pow(10, -6) * h3
        + 1. 793443775071770 * Math.Pow(10, -3) * h2
        - 5. 227303746767450 * Math.Pow(10, -1) * h
        + 1. 100368875131490 * Math.Pow(10, -4) * P3 * h
        + 5. 076769807083600 * Math.Pow(10, -7) * P2 * h2
        + 1. 202580329499520 * Math.Pow(10, -8) * P * h3
        - 7. 278049214744230 * Math.Pow(10, -4) * P2 * h
        - 1. 449198550965620 * Math.Pow(10, -5) * P * h2
        + 5. 716086851760640 * Math.Pow(10, -3) * P * h
        + 5. 818448621582900 * 10
    ;
}

/// <summary>
/// 付録A. 8. 3 式(17)
/// </summary>
/// <param name="P">過冷却液の圧力[MPa]</param>
/// <param name="T">過冷却液の温度[°C]</param>
/// <returns>過冷却液の比エンタルピー[kJ/kg]</returns>
private double f_H_liq(double P, double T)
{
    var K = T + 273.15;
    var K2 = K * K;
    var K3 = K2 * K;

    var P2 = P * P;
    var P3 = P2 * P;

    return
        1. 7902915 * Math.Pow(10, -2) * P3
        + 7. 96830322 * Math.Pow(10, -1) * P2

```

```

+ 5. 985874958 * Math.Pow(10, 1) * P
+ 0 * K3
+ 9. 86677 * Math.Pow(10, -4) * K2
+ 9. 8051677 * Math.Pow(10, -1) * K
- 3. 58645 * Math.Pow(10, -3) * P2 * K
+ 8. 23122 * Math.Pow(10, -4) * P * K2
- 4. 42639115 * Math.Pow(10, -1) * P * K
- 1. 415490404 * Math.Pow(10, 2)
;
}

public override void Commit(SimulationContext context)
{
    //温度, 絶対湿度をリセット

    SensibleHeat = 0.0;
    LatentHeat = 0.0;
    OutTemp = 0.0;
    OutHumidity = 0.0;
    E_AC = 0.0;
    Efficiency = 0.0;
    EtaComp = 0.0;

    //T+1の状態設定
    currentOperation = nextOperation;

    base.Commit(context);
}

public override void Init(SimulationContext context)
{
    //base.Dt = context.AirConditionerDt;
}
}

1.18 ConvectionHeatTransferBlock.cs

using HLFX.Block;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 対流熱伝達カーネル
    /// </summary>
    public class ConvectionHeatTransferBlock : BaseBlock
    {
        /// <summary>
        /// 壁または窓表面
        /// </summary>
        ///private readonly ISurface Surface;

        /// <summary>
        /// 居室または外部
        /// </summary>

```



```

///private readonly ISpaceBlock Space;

private readonly WallFaceConnector WallFace;

public ConvectionHeatTransferBlock(WallFaceConnector
wallface)
{
    WallFace = wallface;
}

public override void Run(SimulationContext context)
{
    //対流熱伝達率[W/m2K]
    //double a_Fc = F.ConvectionHeatTransferRate;
    double alphac = WallFace.AlphaC;

    //壁表面積[m2]
    //double area = WallFace.Area;

    //居室空気温度[°C]
    double spacetemp = WallFace.SpaceTemp;

    //壁面表面温度[°C]
    double surfacetemp = WallFace.SurfaceTemp;

    //対流熱伝達量 (単位[W/m2]) (スペースから面に移動する
    //場合を正、逆なら負)。
    double qc = alphac * (spacetemp - surfacetemp);

    //熱移動
    WallFace.AddHeat(qc);
}
}
}

```

1.19 RadiationHeatTransferBlock.cs

```

using HLFX.Block;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 放射熱伝達ブロック
    /// </summary>
    public class RadiationHeatTransferBlock : BaseBlock
    {
        /// <summary>
        /// 壁面/窓 1
        /// </summary>
        private readonly ILayerBlock R1;

        /// <summary>
        /// 壁面/窓 2
        /// </summary>
        private readonly ILayerBlock R2;

        /// <summary>
        /// 居室の全表面積[m^2]

```

```

/// </summary>
private double A { get; set; }

public RadiationHeatTransferBlock(ILayerBlock src,
ILayerBlock dst, double roomSurfaceArea)
{
    R1 = src;
    R2 = dst;
    A = roomSurfaceArea;
}

public override void Run(SimulationContext context)
{
    //対流熱伝達率[W/m2K] (当面の定数)
    const double a_r12 = 5.0;

    //壁表面積[m2]
    double A_R1 = R1.Area;
    double A_R2 = R2.Area;

    //壁面表面温度[°C]
    double T1 = R1.SurfaceTemp;
    double T2 = R2.SurfaceTemp;

    //対流熱伝達量 (単位[???]) (居室から壁面に移動する
    //場合を正、逆なら負)。
    double q_r1to2 = a_r12 * (A_R1 * A_R2 / A) * (T1 -
    T2);
}
}
}

```

1.20 TotalHeatTransferBlock.cs

```

using HLFX.Block;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 総合熱伝達カーネル
    /// </summary>
    public class TotalHeatTransferBlock : BaseBlock
    {
        private readonly WallFaceConnector WallFace;

        public TotalHeatTransferBlock(WallFaceConnector
wallface)
        {
            WallFace = wallface;
            ///this.src = src;
            ///this.dst = dst;
        }

        public override void Run(SimulationContext context)
        {
            //外部

```



```

//OutsideBlock 0 = src;

//壁面/窓
//ILayerBlock F = dst as ILayerBlock;

//総合熱伝達率[W/m2K]
double alphasat = WallFace.AlphaT;

//放射熱伝達率[W/m2K]
//double a_Fr = F.RadiationHeatTransferRate;

//壁表面積[m2]
//double area = WallFace.Area;

//相当外気温度[°C]
double SAT = WallFace.SpaceTemp;

//壁面表面温度[°C]
double surfacetemp = WallFace.SurfaceTemp;

//総合熱伝達量（単位[W/m2]）（スペースから壁面に移動
//する場合を正、逆なら負）。
double qt = alphasat * (SAT - surfacetemp);

//熱移動
WallFace.AddHeat(qt);
}
}
}

```

1.21 VentilationHeatTransferBlock.cs

```

using HLFX.Block;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 換気熱移動カーネル
    /// </summary>
    public class VentilationHeatTransferBlock : BaseBlock
    {
        /// <summary>
        /// 上流側スペース
        /// </summary>
        private readonly ISpaceBlock UpperSpace;

        /// <summary>
        /// 下流側スペース
        /// </summary>
        private readonly ISpaceBlock LowerSpace;

        /// <summary>
        /// 風量[m3/s]
        /// </summary>
        public double AirVolume { get; set; }

        //public double ExhaustRate { get; set; }

        /// <summary>

```

```

/// スケジュール
/// </summary>
public ScheduleSeries[,] Schedule;

/// <summary>
/// c_air : 空気比熱[J/kgK]。当面、定数とする。
/// </summary>
private const double c_air = 1005;

/// <summary>
/// rho_air : 空気密度[kg/m3]。当面、定数とする。
/// </summary>
private const double rho_air = 1.2;

public VentilationHeatTransferBlock(ISpaceBlock
    upperspace, ISpaceBlock lowerspace, double airvolume,
    ScheduleSeries[,] schedule)
{
    this.UpperSpace = upperspace;
    this.LowerSpace = lowerspace;
    this.AirVolume = airvolume;

    Schedule = new ScheduleSeries[3, 2];

    for (int term = 0; term < 3; term++)
    {
        for (int wd = 0; wd < 2; wd++)
        {
            Schedule[term, wd] = schedule[term, wd];
        }
    }

    ///
    /// 下流の室に登録
    ///
    if (lowerspace is RoomBlock)
    {
        lowerspace.SetVentilation(this);
    }
}

public override void Run(SimulationContext context)
{
    if (LowerSpace is OutsideBlock == false)
    {
        double scvalue = Schedule[context.CurrentTerm,
            context.CurrentWeek].GetScheduleValue(context.Curre
            ntTime);

        double upperspacetemp = UpperSpace.Temp;
        double lowerspacetemp = LowerSpace.Temp;

        double qv = c_air * rho_air * AirVolume * scvalue *
            (upperspacetemp - lowerspacetemp); ///単位[W]

        double upperspacehumidity = UpperSpace.Humidity;
        ///[kg/kg']
        double lowerspacehumidity = LowerSpace.Humidity;
        ///[kg/kg']

        double jv = rho_air * AirVolume * scvalue *
            (upperspacehumidity - lowerspacehumidity); ///単位
            [kg/s]

```

```

        //熱・水分移動(下流側のみに追加)
        LowerSpace.AddVent(qv, jv);
    }
}
}
}
}

```

1.22 ConvectionMoistureTransferBlock.cs

```

using HLFX.Block;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    /// <summary>
    /// 水分伝達カーネル
    /// </summary>
    public class ConvectionMoistureTransferBlock : BaseBlock
    {
        private readonly WallFaceConnector WallFace;

        public
        ConvectionMoistureTransferBlock(WallFaceConnector
        wallFace)
        {
            WallFace = wallFace;
        }

        public override void Run(SimulationContext context)
        {
            //水分伝達率[kg/m2s(kg/kg' )]
            double alphax = WallFace.MoistureTransferRate;

            //壁表面積[m2]
            double area = WallFace.Area;

            //居室絶対湿度[kg/kg']
            double spacehumidity = WallFace.SpaceHumidity;

            //壁面絶対湿度[kg/kg']
            double surfacehumidity = WallFace.SurfaceHumidity;

            //水分伝達量(単位[??]) (居室から壁面に移動する
            場合を正、逆なら負)。
            double J = alphax * area * (spacehumidity-
            surfacehumidity);

            //水分移動
            WallFace.AddMoisture(J);
        }

        public override void Init(SimulationContext context)
        {
            ///base.Dt = context.WallDt;
        }
    }
}

```

```

}
}

```

1.23 Material.cs

```

using HLFX.Block;
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;

namespace HLFX.Utility
{
    /// <summary>
    /// 材料
    /// </summary>
    public class Material
    {
        /// <summary>
        /// 名称
        /// </summary>
        public string Name { get; set; }

        /// <summary>
        /// 熱伝導率[W/mK]
        /// </summary>
        public double 熱伝導率 { get; set; }

        /// <summary>
        /// 空気層熱抵抗[m2K/W] (空気層のみ>0とする)
        /// </summary>
        public double 空気層熱抵抗 { get; set; }

        /// <summary>
        /// 容積比熱[J/m3K]
        /// </summary>
        public double 容積比熱 { get; set; }

        /// <summary>
        /// 密度[kg/m3] (湿気計算材料のみ>0とする)
        /// </summary>
        public double 密度 { get; set; }
    }

    /// <summary>
    /// 材料と厚さの組み合わせ(直列)
    /// </summary>
    public struct WallMaterialList
    {
        public Material Material;
        public double Thickness; // 宮島変更
        Depth→Thickness
    }

    public struct WallSection
    {
        public string SecName; //セク
        ション名
        public double SectionAreaRatio; //面積
        比率
    }
}

```

```

public int MaterialNum;           ///材料数
public List<WallMaterialList> materiallist; ///(材料与厚さ) × 材料数

public WallSection(string name, double ratio, int matnum) : this()
{
    SecName = name;
    SectionAreaRatio = ratio;
    MaterialNum = matnum;
    this.materiallist = new List<WallMaterialList>();
}

///<summary>
///壁仕様(複数断面)
///</summary>
///</summary>
public class WallSpec
{
    ///<summary>
    ///仕様名
    ///</summary>
    public string Name { get; set; }

    ///<summary>
    ///断面数
    ///</summary>
    public int SectionNum { get; set; }

    public List<WallSection> wallsection;

    public WallSpec()
    {
        //Name = name;
        //SectionNum = 0;
        this.wallsection = new List<WallSection>();
    }
}

///<summary>
///窓仕様
///</summary>
public class WindowProperty
{
    ///<summary>
    ///U値
    ///</summary>
    public double UValue { get; set; }

    ///<summary>
    ///ηR値
    ///</summary>
    public double EtaR { get; set; }

    ///<summary>
    ///ηC値
    ///</summary>
    public double EtaC { get; set; }

    ///<summary>
    ///ガラス厚さ
    ///</summary>

```

```

    public double Thickness { get; set; }
}
}

```

1.24 HeatGenerator.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Block
{
    public class HeatGenerator
    {
        private const double EvpHeat = 2510.0;
        ///<summary>
        ///名称
        ///</summary>
        public string Name { get; set; }

        ///<summary>
        ///最大顕熱発熱[W]
        ///</summary>
        public double MaxSensibleHeat { get; set; }

        ///<summary>
        ///最大潜熱発熱[kg/s]
        ///</summary>
        public double MaxVaporGeneration { get; set; }

        ///<summary>
        ///対流成分比率[-]
        ///</summary>
        public double ConvectionRate { get; set; }

        ///<summary>
        ///スケジュール
        ///</summary>
        public ScheduleSeries[,] HeatGenSchedule;

        ///<summary>
        ///コンストラクタ
        ///</summary>
        public HeatGenerator(string name, double maxsensibleheat, double maxvapor, double convectionrate, ScheduleSeries[,] schedule, IList<RoomBlock> applyrooms)
        {
            //string[,] ScheduleID = new string[3, 2];
            Name = name;
            MaxSensibleHeat = maxsensibleheat;
            MaxVaporGeneration = maxvapor;
            ConvectionRate = convectionrate;

            HeatGenSchedule = new ScheduleSeries[3, 2];

            for (int term = 0; term < 3; term++)

```

```

    {
        for (int wd = 0; wd < 2; wd++)
        {
            HeatGenSchedule[term, wd] = schedule[term, wd];
        }
    }

    for (int i = 0; i < applyrooms.Count; i++)
    {
        RoomBlock applyroom = applyrooms[i];
        applyroom.SetHeatGenerator(this);
    }
}

/// <summary>
/// ある時点(期,曜日,時刻,分)を受け取り,顕熱発熱量・潜
/// 熱発熱量を返す
/// <param name="time"></param> 時刻・分を4桁整数化し
/// たもの
/// <param name="qgenr"></param> 発熱量(対流成分) [W]
/// <param name="qgenc"></param> 発熱量(放射成分) [W]
/// <param name="jgen"></param> 水分量[kg/s]
/// </summary>
public void GetHeat(SimulationContext context, out
double qgenc, out double qgenr, out double jgen)
{
    double scvalue = HeatGenSchedule[context.CurrentTerm,
context.CurrentWeek].GetScheduleValue(context.Current
Time);

    qgenr = this.MaxSensibleHeat * (1.0 - ConvectionRate)
* scvalue;
    qgenc = this.MaxSensibleHeat * ConvectionRate *
scvalue;
    jgen = this.MaxVaporGeneration * scvalue;    ///[kg/s]
}

}

public struct ScheduleUnit
{
    /// <summary>
    /// スケジュールユニット(開始時刻と値の組み合わせ)
    /// </summary>

    public int StartTime;           ///開始時刻 12時30分
    =1230
    public double ScheduleValue;    ///値
    ///終了時刻はない(次の配列の開始時刻を見る)
}

public class ScheduleSeries
{
    List<ScheduleUnit> ScheduleSet;
    //public ScheduleUnit[] ScheduleSet { get; set; }

    /// <summary>
    /// コンストラクタ
    /// </summary>
    /// <param name="scheduleunits"></param>
    public ScheduleSeries()
    {
        ScheduleSet = new List<ScheduleUnit>();
    }

    /// <summary>

```

```

/// スケジュールユニット(開始時刻と値のセット)を追加す
/// る
/// </summary>
/// <param name="scheduleunits"></param>
public void UnitAdd(int starttime, double scvalue)
{
    ScheduleSet.Add(new ScheduleUnit { StartTime =
starttime, ScheduleValue = scvalue });
}

/// <summary>
/// スケジュール値を返す
/// </summary>
/// <param name="scheduleunits"></param>
public double GetScheduleValue(int time)
{
    for (int i = 0; i < ScheduleSet.Count - 1; i++)
    {
        if (time < ScheduleSet[i + 1].StartTime)    ///time
        が次のセットの開始時刻より早くなったところで確定
        return ScheduleSet[i].ScheduleValue;
    }

    ///ここまでで見つからなかった場合,最後のセットが適用
    される
    int j = ScheduleSet.Count - 1;
    return ScheduleSet[j].ScheduleValue;
}

}
}

```

1.25 SMASHWeatherLoader.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;

namespace HLFX.Utility
{
    /// <summary>
    /// SMASH形式読込データ
    /// </summary>
    public struct SMASHWeatherData
    {
        /// <summary>
        /// 気温 [°C]
        /// </summary>
        public double Temperature;

        /// <summary>
        /// 絶対湿度 [kg/kg']
        /// </summary>
        public double Humidity;

        /// <summary>
        /// 法線面直達日射量 [W/m2]
        /// </summary>
        public double Ids;

        /// <summary>
        /// 水平面天空日射量 [W/m2]
        /// </summary>
        public double Iis;
    }
}

```

```

/// <summary>
/// 水平面夜間放射量 [W/m2]
/// </summary>
public double Ins;

/// <summary>
/// Sin(太陽高度)
/// </summary>
public double SinH;

/// <summary>
/// Cos(太陽高度)
/// </summary>
public double CosH;

/// <summary>
/// Sin(太陽方位角)
/// </summary>
public double SinA;

/// <summary>
/// Cos(太陽方位角)
/// </summary>
public double CosA;

/// 曜日は取得しない
}

public static class SMASHWeatherLoader
{
/// <summary>
/// 各月日数
/// </summary>
private static readonly int[] cMonth = new int[] { 31,
28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

/// <summary>
/// kcal/h → Wの換算係数
/// </summary>
public const double kcaltoW = 1.163;

/// <summary>
/// 月、日、時刻を行番号に変換します。
/// SMASH形式想定、時刻は1～24を指定
/// </summary>
/// <param name="m"></param>
/// <param name="d"></param>
/// <param name="h"></param>
public static int GetIndex(int m, int d, int h)
{
int lDay = d - 1;
for (int i = 0; i < m - 1; i++)
{
lDay += cMonth[i];
}
lDay *= 24;

var ret = lDay + h - 1; //1時スタート

if (ret == -1) return 8759; //1月1日0時=>8759
return ret;
}

/// <summary>
/// ファイル読込
/// </summary>

```

```

/// <param name="fileName">ファイル名</param>
public static IEnumerable<SMASHWeatherData>
ReadData(string fileName)
{
using (StreamReader reader = new
StreamReader(fileName, Encoding.ASCII))
{
int lLineCount = 1;
string lLine = "";
SMASHWeatherData data;

//パース処理用補助関数
Func<string, int, int, int> GetValue = (label, a,
b) =>
{
int lValue;
if (int.TryParse(lLine.Substring(a - 1, b - a +
1).Trim(), out lValue) == false)
{
throw new Exception(string.Format("{0}行目: {1}
のデータが数値に変換出来ません。", lLineCount,
label));
}
return lValue;
};

//地点番号を無視
reader.ReadLine();
lLineCount++;

//25行ごとに365日
for (int d = 0; d < 365; d++)
{
//一行無視
reader.ReadLine();
lLineCount++;

for (int h = 0; h < 24; h++, lLineCount++)
{
#region 一行読込

lLine = reader.ReadLine();

//気温
//1～4文字目: 気温。10で除すことで単位[°C]となる。
data.Temperature = GetValue("気温", 1, 4) / 10.0;

//絶対湿度
//5～7文字目: 絶対湿度。10,000で除すことで単位
[kg/kg' ]となる。
data.Humidity = GetValue("絶対湿度", 5, 7) /
10000.0;

//法線面直達日射量
//8～10文字目: 法線面直達日射量。単位[kcal/m2h]
data.Ids = GetValue("法線面直達日射量", 8, 10) *
kcaltoW;

//水平面天空日射量
//11～13文字目: 水平面天空日射量。単位[kcal/m2h]
data.Iis = GetValue("水平面天空日射量", 11, 13) *
kcaltoW;

//水平面夜間放射量
//14～16文字目: 水平面夜間放射量。単位[kcal/m2h]
data.Ins = GetValue("水平面夜間放射量", 14, 16) *

```

```

kcaltoW;

//太陽高度Hの正弦 (sinH)
//17~19文字目: 太陽高度Hの正弦 (sinH)。1,000で
除す必要がある。
data.SinH = GetValue("太陽高度Hの正弦 (sinH) ",
17, 19) / 1000.0;

//太陽高度Hの余弦 (cosH)
//20~22文字目: 太陽高度Hの余弦 (cosH)。1,000で
除す必要がある。
data.CosH = GetValue("太陽高度Hの余弦 (cosH) ",
20, 22) / 1000.0;

//太陽方位角Aの正弦 (sinA)
//23~26文字目: 太陽方位角Aの正弦 (sinA)。1,000
で除す必要がある。太陽が真南より東側にある場合は
数値の前に"-"が付く。
data.SinA = GetValue("太陽方位角Aの正弦 (sinA) ",
23, 26) / 1000.0;

//太陽方位角Aの余弦 (cosA)
//27~30文字目: 太陽方位角Aの余弦 (cosA)。1,000
で除す必要がある。太陽が真東・真西より北側にある
場合は数値の前に"-"が付く。
data.CosA = GetValue("太陽方位角Aの余弦 (cosA) ",
27, 30) / 1000.0;

#endregion

yield return data;
}
}

reader.Close();
}

/// <summary>
/// ファイル読み
/// </summary>
/// <param name="fileName">ファイル名</param>
public static IList<SMASHWeatherData>
ReadAllIData(string fileName)
{
return ReadData(fileName).ToList();
}

/// <summary>
/// 分刻みの補間値を返します。
/// </summary>
/// <param name="m">月</param>
/// <param name="d">日</param>
/// <param name="h">時刻</param>
/// <param name="M">分</param>
public static SMASHWeatherData
GetLinearInterpolated(int m, int d, int h, int M,
IList<SMASHWeatherData> data)
{

//経過時間
int index_prev = GetIndex(m, d, h); //1月1日1時を0
としたインデックス
int index_next = index_prev + 1; //次の時刻のイ
ンデックス

//補間値

```

```

if (index_prev == 8759) //12月31日24時
の次の時刻
{
index_next = 0;
}
double a = M / 60.0;
var ret = new SMASHWeatherData
{
Temperature = (data[index_next].Temperature -
data[index_prev].Temperature) * a +
data[index_prev].Temperature,
Humidity = (data[index_next].Humidity -
data[index_prev].Humidity) * a +
data[index_prev].Humidity,
Ids = (data[index_next].Ids - data[index_prev].Ids)
* a + data[index_prev].Ids,
Iis = (data[index_next].Iis - data[index_prev].Iis)
* a + data[index_prev].Iis,
Ins = (data[index_next].Ins - data[index_prev].Ins)
* a + data[index_prev].Ins,
SinH = (data[index_next].SinH -
data[index_prev].SinH) * a + data[index_prev].SinH,
CosH = (data[index_next].CosH -
data[index_prev].CosH) * a + data[index_prev].CosH,
SinA = (data[index_next].SinA -
data[index_prev].SinA) * a + data[index_prev].SinA,
CosA = (data[index_next].CosA -
data[index_prev].CosA) * a + data[index_prev].CosA,
};

//sin・cosの調整
double b;
b = ret.SinH * ret.SinH + ret.CosH * ret.CosH;
if (b > 0.0)
{
ret.SinH /= Math.Sqrt(b);
ret.CosH /= Math.Sqrt(b);
}

b = ret.SinA * ret.SinA + ret.CosA * ret.CosA;
if (b > 0.0)
{
ret.SinA /= Math.Sqrt(b);
ret.CosA /= Math.Sqrt(b);
}

return ret;
}
}

```

1.26 Util.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HLFX.Utility
{
public static class Util
{

```

```

public const double AbsoluteTemperatureConstant =
273.16;

/// <summary>
/// 温度[°C]から絶対温度[K]への変換
/// </summary>
/// <param name="T">温度[°C]</param>
/// <returns>絶対温度[K]</returns>
public static double ToAbsoluteTemperature(double T)
{
    double Tk = T + AbsoluteTemperatureConstant;

    return Tk;
}

/// <summary>
/// 絶対温度[K]から温度[°C]への変換
/// </summary>
/// <param name="Tk">絶対温度[K]</param>
/// <returns>温度[°C]</returns>
public static double ToRelativeTemperature(double Tk)
{
    double T = Tk - AbsoluteTemperatureConstant;

    return T;
}

/// <summary>
/// 温度T[°C]の空気の飽和水蒸気圧[Pa]を求める。
/// </summary>
/// <param name="T">絶対温度[°C]</param>
/// <returns>温度T[°C]の空気の飽和水蒸気圧
[Pa]</returns>
public static double GetSaturatedVaporPressure(double
Tk)
{
    const double a1 = -6096.9385;
    const double a2 = 21.2409642;
    const double a3 = -2.711193 * 0.01;
    const double a4 = 1.673952 * 0.00001;
    const double a5 = 2.433502;

    double Ps = Math.Exp(a1 / Tk + a2 + a3 * Tk + a4 *
Tk * Tk + a5 * Math.Log(Tk));

    return Ps;
}

/// <summary>
/// 水蒸気圧[Pa]から絶対湿度[kg/kg']を求める。
/// </summary>
/// <param name="p">水蒸気圧[Pa]</param>
/// <returns>絶対湿度[kg/kg']</returns>
public static double GetAtmAbsolute(double p)
{
    const double Pa = 101325; //大気圧

    double X = (0.622 * p) / (Pa - p);

    return X;
}

/// <summary>
/// 温度と絶対湿度から相対湿度を求める。
/// </summary>
/// <param name="Tk">絶対温度[K]</param>
/// <param name="X">絶対湿度[kg/kg']</param>

```

```

/// <returns>相対湿度[%]</returns>
public static double GetRelativeHumidity(double Tk,
double X)
{
    //温度Tから飽和絶対湿度Xsを計算
    double Ps = GetSaturatedVaporPressure(Tk);
    double Xs = GetAtmAbsolute(Ps);

    //飽和絶対湿度Xsと絶対湿度Xから相対湿度を求める
    double h = X / Xs * 100.0;

    return h;
}

/// <summary>
/// 温度と絶対湿度から相対湿度, または温度と相対湿度か
ら絶対湿度を求める。
/// Wexler-Hyland式
/// </summary>
/// <param name="temp">温度[°C]</param>
/// <param name="humidity">絶対湿度[kg/kg']または相対
湿度[-]</param>
/// <param name="calctype">1:絶湿計算, 2:相湿計算, 3:
相湿計算(100%超を認める)</param>
/// <returns>相対湿度[%]</returns>
public static double CalcHumidity(double temp, double
humidity, int calctype)
{
    const double C1 = -5800.2206;
    const double C2 = 1.3914993;
    const double C3 = -0.048640239;
    const double C4 = 0.000041764768;
    const double C5 = -0.000000014452093;
    const double C6 = 6.5459673;

    double abstemp = temp + AbsoluteTemperatureConstant;

    double matpress = C1 / abstemp + C2 + abstemp * (C3 +
abstemp * (C4 + abstemp * C5))
        + C6 * Math.Log(abstemp);

    matpress = Math.Exp(matpress); ///[Pa]

    double matabshumidity = 0.62198 * matpress /
(101325.0 - matpress); ///[kg/kg']

    if (calctype == 1) //絶湿を計算
        return matabshumidity * humidity; //humidityは
相対湿度
    else //相湿を計算
    {
        double rh = humidity / matabshumidity;
        //humidityは絶対湿度

        if (calctype == 3) //calctype=3のとき
            return rh; //は1超でもそのまま返す
        else
        {
            if (rh > 1.0) rh = 1.0;
            return rh;
        }
    }
}
}
}

```

© 建築研究資料 第 190 号

平成 31 年 4 月 印刷・発行
編集・発行 国立研究開発法人建築研究所

本資料の転載・複写の問い合わせは下記まで
国立研究開発法人建築研究所企画部企画調査課
〒305-0802 茨城県つくば市立原 1 番地
電話 (029) 864-2151(代)