修士論文

植物工場における効率的な栽培レシピ探索のためのデータ解析基盤の実装

2018年2月1日

指導教員　川原　圭博　准教授

東京大学大学院情報理工学系研究科
電子情報学専攻　48-166404

飯塚　達哉

内容梗概

完全制御型の植物工場では膨大なイニシャルコスト及びランニングコストが生じるために，生育速度などを指標とした生産性を高める必要がある．生産性は照明時間や気温など栽培環境制御則「レシピ」に大きく依存するため，最適なレシピの発見が重要となってくる．従来の開発は，植物の生態生理学の研究によって得られた知見から最適レシピにあたりをつけ，多くの試行錯誤によって生育を早める環境条件を見つけ出すという現場の研究開発者の経験と勘に大きく依存したものであったが，これはヒューリスティックな手法であり拡張性が低い．特定の設備下で特定の品種にのみ最適なレシピを発見したとしても，設備や対象とする品種が変更された場合にはそれまでノウハウの転用が難しく，再び試行錯誤によるレシピの改良を強いられる．そこで，植物工場内に設置したセンサデータからレシピの良し悪しを定量的に評価し改善することができれば，設備や品種に依存しないレシピ改良手法の確立が可能となる．

過去にも，リカレントニューラルネットワーク (RNN) などの表現力の高い時系列モデリングが可能な手法を用い，植物の環境応答モデルを構築してから所望の環境応答を得るレシピを逆算するという研究が行われてきた．しかし，これは予測モデルがレシピの探索範囲で常に高い精度をもつことで初めて有効となる手法であるが，RNN などの自由度の高い時系列モデルなどを用いる場合は，学習するべきレシピの組み合わせの数が膨大となってしまい，変化の遅さのためデータの収集コストの高い植物には不適である．

また，多くの研究は時系列データの収集に関する方法論は考察されておらず，既にデータを持っていることを前提にしているが，所望の環境に制御可能な植物工場においては，各時刻での目標環境値を「賢く」選択し効率的にレシピを評価・改善していくことが重要になってくる．ここで述べる「賢い」目標環境値の選択とは，これまでのサンプルからでは予測精度が低いレシピを選択する「探索」と 生産性が高いと予想されるレシピの周辺を重点的に選択 する「活用」の両点を考慮する選択のことである．

そこで本稿では，葉面積の即日の成長率がコンテキスト依存せずレシピによって一定であるという単純なモデルを仮定することで，レシピを稼働したときの報酬が即日で得られるバンディット問題とみなせるようにし，ベイズ最適化を用いた探索と活用のトレードオフを考慮したレシピ探索手法を提案した．1 日ごとにレシピを変更していったときの即日の成長率を観測し，レシピと成長率の対応関係を考察し，バンディット問題への帰結可能性を示した．なお実験環境として，MIT Media Lab の Open Agriculture Intitiative (OpenAg) というチームが作成したオープンソースな植物工場である Personal Food Computer を作成し，使用したため，その実装の詳細も本稿で述べた．

目 次

図 目 次

表 目 次

第1章

序論

1.1　本研究の背景

近年，農業に ICT を活用する取り組みが注目され，多様なセンサデータの解析や作業支援ツールによって生産性を高めることが期待されている．特に，高度な環境制御技術と環境センシング技術を用いて生産性の高い植物生育を実現する植物工場は，2009 年の農商工連携研究会報告書 [4] においてその重要性が強調され，事業として植物工場を営む企業が増えた．日本国内において照明に人工光を用いた植物工場（完全制御型植物工場）の施設数は，2011 年の 64 箇所から 2016 年 2 月時点には 191 箇所にまで増加した [5]．しかし一方で，完全制御型植物工場で植物を生産して黒字化を達成している企業は全体の 20.6%と一握りに限られる [6]．利益が生じにくい主な要因として，野菜の単価が小さいことや植物工場のランニングコストが非常に大きいことが挙げられる．黒字化を達成するためには，植物工場の各時刻における制御信号もしくは環境の目標状態を定めた栽培環境制御則「レシピ」の生産性を高める必要がある．株式会社スプレッドは長年の研究開発によって独自のレシピを確立し，レタスの栽培において 97%という高い歩留まり率と日産 2 万株以上の生産量を誇る生産性を可能とした [7].

このように植物工場事業の収益は運用するレシピの生産性に大きく影響するにも関わらず，未だにレシピの効率的な開発手法は確立されていない．多くの植物工場におけるレシピの開発法は，植物の生態生理学の研究によって得られた知見から最適レシピにあたりをつけ，多くの試行錯誤によって生育を早める環境条件を見つけ出すという現場の研究開発者の経験と勘に大きく依存したものである．この方法論は多くの時間を要するノウハウの確立が必要となり，人件費や時間などのコストが高くなる．また，植物の生育に影響を与える環境要因は日照時間や温度，湿度，CO2 濃度，水温，pH，液肥の投与量など多様であり，これらが互いに干渉する複合環境制御においては，各環境要因に対して局所的に解明されている植物のダイナミクスの知見を利用して複合最適なレシピを発見することは難しい．

そこで，生育期間中に収集される多様な環境センサデータに機械学習を適用することで最適なレシピを得ようという研究が過去に多く行われてきた．多様なセンサデータを用いて，植物を取り巻く環境に関するデータと環境応答に関するデータを測定し，環境値の入力に対して環境応答を出力としたブラックボックスをモデル化することで，最適な環境応答を得る入力を逆算するという提案が行われている．予測モデル構築の手法として，時系列のモデルを表現できるリカレントニューラルネットワークを用いて，各時刻の環境値が与えられたときの植物の環境応答が予測可能となっている [2]．たしかに，リカレントニューラルネットワークは時系列データのモデリングにおいて高い表現力を持ち有効な手法の一つである．しかし，高い表現力を持つ機械学習モデルを採用するほど，植物の環境応答をモデル化するために必要な学習データの数は多くなり，データの収集コストが大きい植物には向かない．近年注目を浴びている深層学習などのビッグデータを用いた機械学習手法は，大量のデータが既にあることを前提としているため，植物への適用は困難である．高価なカメラを用いて分解能を高めることで植物の小さな変化も観測し，短い期間でより多くのデータを収集してデータ解析を行う High Throughput Phenotyping という分野もあるが，高価な測定器を用いるためにその応用先は主に育種などのより付加価値が高いものとなっている [8, 9, 10].

そこで，植物の成長におけるデータ解析・機械学習手法の適用の困難さを考慮した，植

物の生育を早めるレシピを如何に短期間で発見するかという実用的なレシピ探索手法の開発が必要とされている．本稿では，植物工場での実験環境が与えられた時に，どのように環境を制御して，どのようなデータを集めれば最適なレシピに早く到達できるかということを検討する．

1.2　本研究の目的

本研究の目的は，最適なレシピを早く発見するデータドリブンな手法の確立である．一般的に植物を対象としたデータ解析において以下の困難さが挙げられている．

- 植物の生育速度が遅いため，有意な変化を捉えたデータの収集コストが大きい．
- 植物の成長は一方向性を持つため，一回の生育において特定の状態は一度しか観測できない．
- 植物の環境応答は様々な環境要因によって決定され，レシピを決定する環境要因の組み合わせは膨大である．

これらの困難を乗り越えて，現実時間で実行可能な最適レシピの探索手法を検討する．植物工場におけるデータを収集する流れは,「環境の制御」→「植物の環境応答の観測」の繰り返しである．従来の手法では多くの時間を要するこの試行錯誤のサイクルをどれだけ短く・効率的にすることができるか，ということが本研究のチャレンジである．サイクルを短くするために，本稿では植物の葉面積の日毎の成長率を観測し，レシピの良し悪しを即座に反映する指標である環境応答として採用する．また，次の制御点を賢く選択することでサイクルを効率的に回すための手法として，日毎の成長率を最大化するレシピを選択するバンディット問題に帰結させることを提案する．具体的にはベイズ最適化を適用することにより「探索」と「活用」のトレードオフを考慮したベイズ最適な選択が可能となる．

また，この提案を検証するにあたり植物工場を再現する実験環境が必要となる．MIT Media Lab の Open Agriculture Intitiative というチームがオープンソースとして作成可能な植物工場というコンセプトで Personal Food Computer(PFC) を作成し，その設計を公開している．本研究では，この PFC を実装し，実験環境として使用する．そしてレシピ探索のために必要となるデータ解析基盤のソフトウェア実装も行い，データ収集基盤及び解析基盤がレシピ探索のために備えているべき要件を明らかにする．

1.3　本研究の寄与

本研究の寄与は以下にまとめられる．

- 簡易型の植物工場においてデータ収集及び解析の基盤を実装し，レシピ探索において必要なシステム要件を述べた．
- 植物の環境応答の指標として成長率を取り上げ，時間変化しない単純な環境応答モデルを検討した．

- 提案した環境応答モデルにおいて，探索と活用を考慮したレシピ探索手法を提案した．

1.4　本論文の構成

続く 2 章では，植物工場におけるデータドリブンでのレシピ最適化手法に関して今まで行われてきた研究をまとめる．そこで，従来の手法は最適なレシピを探索するというよりは，高い表現量を持った植物の環境応答のモデリングを行うことに注力されてきたことを述べる．3 章では，本提案を検証するための実験環境として作成した簡易型植物工場 Personal Food Computer (PFC) の実装について詳細を述べる．4 で，レシピ探索のために必要なデータ解析の要件を述べ，PFC 上で得られたデータを解析するために実装したシステム概要を記す．5 では，実装した PFC 及びデータ収集基盤を用いてベイズ最適化を適用したレシピ探索手法について述べる．そして最後に本稿のまとめとして 6 章を記す．

本稿は以下の各章により構成される．

第 1 章 序章

第 2 章 植物工場におけるレシピ最適化の関連研究

第 3 章 Personal Food Computer を用いたデータ収集基盤の実装

第 4 章 レシピ探索のためのデータ解析基盤の実装

第 5 章 成長率モデルを利用したベイズ最適なレシピ探索手法

第 6 章 結論

第2章

植物工場におけるレシピ最適化の関連研究

2.1　概要

本研究は，植物工場において最適なレシピを素早く発見するための方法論を確立することを目的としている．これまでにも，植物の生育を早めるために環境をどのように制御すればよいかに関する研究は多く行われてきた．しかし，この問題は植物の生育に由来する以下の特徴を持っており，未だに決定的な手法は確立されていない．

- 植物の成長および環境応答は遅く，変化を観測して有意義なデータを取得するのに多くの時間を要する．
- 様々な環境要因が植物の生育に変化を与えるため，複合環境下での植物の生育はブラックボックスである．
- レシピは多種類の環境要素で構成されており，その探索空間は膨大である．
- 植物の体は時間とともに積算的に形成されるため，生育中の全期間を通して環境が植物に与える影響を考慮する必要がある．

この分野に関する今までの研究で用いられる手法のほとんどは，植物の生態生理学の知見を利用した方法と取得可能なセンサ値を基にデータドリブンで制御則を決定する方法の二つに分類できる．前者を採用した研究を 2.2 章でまとめて述べる．しかし，これらの方法は照明や温度，CO2 濃度など，1 種類もしくは数種類の環境要因を対象とした環境の最適化を用いたい場合に有効であることは示されているが，環境はより多種類の環境要因で構成されるため，これらを全て考慮したレシピの最適化は難しい．そこで，本研究では後者のアプローチを取った．データドリブンで制御則を決定する方法もまた 2 種類に分けられる．一つ目は，稼働しているレシピに対して植物の環境応答を予測する機械学習モデルを構築した後に，そのモデルを用いて所望の環境応答を得る入力（レシピ）を求めるという手法である．この詳細を 2.3 に示す．二つ目は，環境応答の予測モデルの構築と次にサンプルするレシピの選択を同時に行う手法である．強化学習を用いたものなどがこの例にあたり，2.4 節に示す．

2.2　植物の生態生理学の知見を利用した最適環境制御

植物の生育において最も重要な活動の一つが光合成である．光合成は葉の葉緑体の内部で行われ，光のエネルギーを用いて二酸化炭素と水からを炭水化物を生成することで身体を大きくする．一般的に植物工場において消費される総電気量のうち照明に用いられる割合は大きいため，効率的な照明の制御は重要である．一方で，光が強すぎると植物は全てのエネルギーを光合成には使用できず，余ったエネルギーを熱として植物体から放射されてしまい，LED のエネルギーは無駄になってしまうという植物の特性が知られており，適切な光量がますます重要になる．そこで，Van らは葉の電子伝達速度をリアルタイムで観測しながら栽培することで，光合成使用可能光強度を推定し，それによって LED の照射強度を調整するシステムを開発した [1]．システムの外観を図 2.1 に示す．電子伝達速度を測定することで，光合成に使用可能な光量が分かる．これを計測しながら照明の制御を行うことで，植物の成長に寄与しない無駄となる光の量を減らせる．本システムによって吸

収されない LED のエネルギーを最小化する制御が可能となり，エネルギー利用効率を高めた．

図 2.1: バイオフィードバック機構が備わった LED 照射システムの外観 [1]．葉の電子伝達速度をリアルタイムで計測し，計測値をもとに LED の照射強度及び照射時間を決定することで LED の消費電力の利用効率が高まる．

2.3　環境応答のモデル化を基にした手法

最適レシピを求めるためには，様々なレシピを試し，成功例と失敗例に関する大量データの蓄積が必須である．しかし，種植えを始めてから収穫するまで短い植物でも数週間，長い植物だと数年以上という時間が必要となる．つまり，1 つの正解データを得るのに最短でも数週間以上かかる．さらに，土壌栽培の場合は土壌によって生育状況が異なり，またその原因が未だにわかっていない．よって，ある土壌で収集したデータを別の土壌に容易には適用できない．

この現状において，植物の成長期機構をモデル化するための技術として Speaking Plant Approach (SPA) がある．SPA 技術は，植物生体情報を計測し，それに基づいて栽培環境を最適に制御する一連の技術である．SPA コンセプト [11, 12, 13] が提唱されたのは 30 年前に遡るが，近年の様々なセンサの開発及び高性能化・低コスト化によっていよいよ実用化されている技術である．この SPA のコンセプトに基づいた研究としては 20 年ほど前から行われてきた．竹内らはルッコラ栽培において背丈の値をロジスティック回帰を適用することで，高い精度で背丈の成長モデルを構築することに成功している [14]．平藤らは，環境応答として光合成速度及び茎長を周期的に計測し，時系列データに対応可能なニューラルネットワークを用いて，環境応答の予測モデルを構築した [2]．観測されたデータを図 2.2 に示す．このグラフにおける光合成速度は CO2 濃度の減少速度をもとに計算された．

閉空間における栽培実験であるため，CO2 濃度の減少は光合成速度と高い相関を持っていることが考えられるためである．図 2.3 は学習したニューラルネットワークを用いて光合成速度の予測値と実測値を示したものである．高い精度で予測を行えていることが分かる．

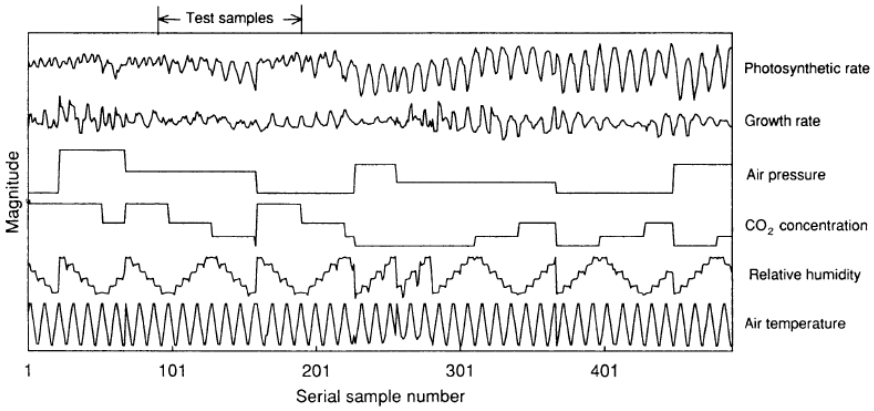

図 2.2: 環境値と光合成速度の時系列データの関係性を示したグラフ [2].

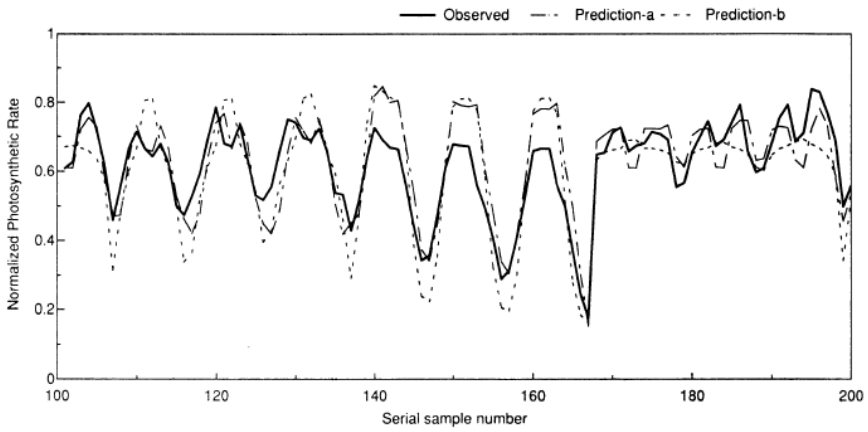

図 2.3: 各時刻の植物の光合成速度に関してリカレントニューラルネットワークを用いた予測値と実測値を比較したグラフ [2].

この予測器を用いてあるレシピを入力したとき，環境応答を出力として得ることができる．つまり，特定のレシピを実行したときに所望の結果を得られるかどうかがシミュレーション可能となった．このようにして，学習したニューラルネットワークをシミュレータとして，遺伝的アルゴリズム (GA) を用いてレシピを最適化する手法が提案されている [15, 16, 17, 18, 19].

Yumeina らは，環境に関するセンサデータからトマトの成長率を出力するようなニューラルネットワークを学習させた後に，ニューラルネットワークの適用に GA を用いて次のステップで最適なレシピを選択する手法を提案した．そのシステムのブロックダイアグラムを図 2.4 に示す．ニューラルネットワークによる予測と GA を用いた最適化を組み合わ

せることによって，トマトの成長率を高めることがこの研究で報告されている．また，植物の環境応答をモデル化するためにニューラルネットワークを用いる手法を多く利用されている [20, 21, 22].

しかしこれらの手法は，ニューラルネットワークがどのような環境値に対しても高い精度で環境応答を予測するという前提を置いている．機械学習に於いて一般的に，入力となる環境値が学習に用いられたデータと近い場合には予測精度は高いが，学習データと大きく異なる場合には予測精度は担保できない．さらにニューラルネットワークでは出力した予測値の確度を定量的に示すことは難しく，その予測がどの程度の確率で正しいのか判断できない状態で GA の計算を進めてしまう．この手法を用いる場合，ニューラルネットワークの学習に用いるデータセットが大きく，様々なバリエーションを含むことが望ましい．植物の生育実験で得られるデータであるため，大規模なデータセットを得ることは難しく，どのように環境を制御して学習データを集めることが重要となるが，これらの研究ではデータのサンプリング方法を対象とはしていない．次節では，どのように環境を制御して学習データを集めるか，ということまで考慮した強化学習を適用した手法について述べる．

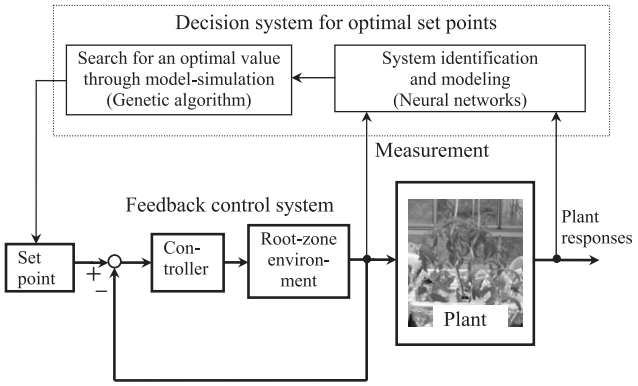

図 2.4: トマトの水耕栽培における SPA に基づいた最適環境制御システムのブロックダイアグラム 2.4.

2.4　サンプリングとモデル化を同時に行う手法

植物工場においてデータドリブンで植物の環境応答モデルを構築する際には，どのようにして学習データを集めるか，どのようにして環境制御値を決定していくかが非常に重要である．この問題を，植物の状態が制御環境によって変化し，観測可能な成長量を最大化するように環境を制御する問題と考えれば，強化学習の枠組みを利用可能である．若原らは，葉大根の伸長の成長量を報酬とした強化学習を適用した [3]．閉空間での生育を行い，伸長が最大となるように投与する液肥の組成を最適化する問題に着手した．ここで，投与する液肥の組成を強化学習における行動，葉大根の伸長を状態，成長率を報酬としている．通常の強化学習を適用する際に生じる弊害を次のようにまとめた．

- 報酬が長遅延で生じること.
- 行動と観測のタイミングが異なること.
- 一回の生育において同じ状態は一度しか観測できない.

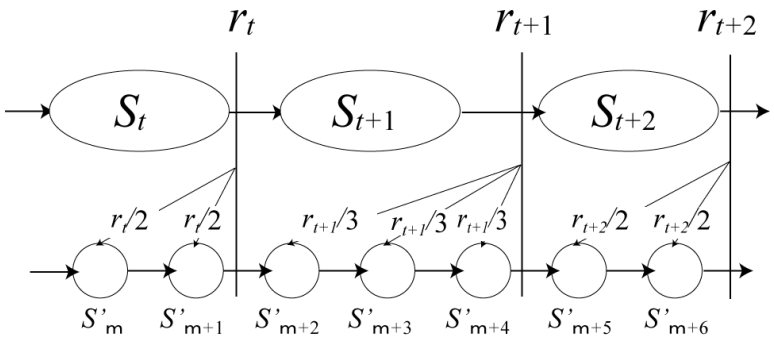

図 2.5: 若原の提案方式における制御上学習系列に対する報酬の与えられ方 [3].

そこで，図 2.5 に示す，報酬が長遅延で得られる状況における強化学習のモデルを新たに提案した．上段は測定された伸長値の状態遷移を示し，下段は液肥投与による状態遷移を示している．この状態遷移モデルでは，得られた報酬は同じ状態 S_t に属する行動 S'_m に均等に配分される．この報酬モデルを基に Q 学習を適用して栽培実験を行った．その実験結果を 2.6 に示す．結果として，強化学習によって成長率が改善されていないことが分かる．試行回数が少ないために学習が不充分であることが最大の原因と考えられる．若原は，は大根の成長に関するシミュレーションを行い，従来の強化学習よりも早く収束することを示している．

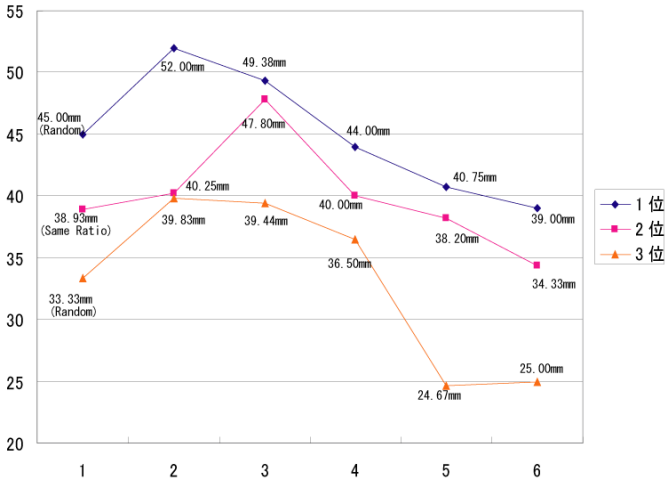

図 2.6: 強化学習を適用したときの葉大根の成長の伸長の時間遷移グラフ [3].

第3章

Personal Food Computer を用いたデータ収集基盤の実装

3.1　概要

本章では，植物工場におけるレシピ探索を再現するために構築した実験環境について詳述する．我々は MIT Media Lab の Open Agriculture Intitiative (OpenAg) というチームがオープンソースとして設計を公開している Personal Food Computer (PFC) を組み立て，そこで様々な環境データ及び環境応答を計測しながら植物の栽培を行った．本章の構成として，はじめに一般的な植物工場について特徴及び備えている機能を述べる．次に PFC のハードウェア構成や作成方法を 3.3 節，ソフトウェア構成を 3.4 節において，それぞれ OpenAg が公開していない情報を重点的に述べる．最後に，我々が PFC とは別に用意した育苗環境について，PFC への定植までの流れとともに述べる．

3.2　簡易植物工場 Personal Food Computer

3.2.1　植物工場の定義

植物工場とは野菜や苗を中心とした作物を施設内で光や温湿度，二酸化炭素濃度，培養液などの環境条件を人為的に制御することで，栽培に対する季節や場所の影響を少なくして安定で高効率な生産を行うシステムを表す．特に完全制御型植物工場は光源に人工光のみを用いたものをさす．露地栽培や温室栽培と比較するとランニングコストとなる電気代およびイニシャルコストとなる工場建設費は共に大きい．そのため，その高いコストに見合うだけの生産性が求められる．高い生産性は歩留まり率の上昇や成長速度の向上などの単位時間あたりの収穫量の工場，または低カリウム含有や無菌栽培などの植物工場独自の付加価値を持つことで達成されることが多い．以下では，植物工場で用いられている主要技術について述べる．

3.2.2　システム構成及び環境制御技術

植物工場は施設や環境制御機器，栽培装置，各種センサ，栽培のノウハウなどを含めた総合的なシステム技術である．完全制御型植物工場では，外界の光と熱を遮断するために天井や壁を断熱性の高い材料でつくる．この外部環境から隔離された閉鎖空間の環境制御技術が肝となる．以下に各種の環境制御技術について概要を記す．

水耕栽培技術

ほとんどの植物工場は生育速度を高めるために水耕栽培を採用している．水耕栽培は根が陰陽に乖離した無機イオンを吸収することに着目し，無機化学肥料を水に溶かして陰陽イオンに乖離させ，根から直接吸収させる方法である．微生物の働きが少ないので肥料がすぐに効く一方で，土壌栽培では得られる緩衝作用がほとんどない．そのため，肥料成分の投与が不適切であったり，水中の溶存酸素量が少なかったりすると生育障害が生じやすい．

そこで，肥料の成分組成や濃度（電気伝導度 EC），培養液の pH，水量，水温，溶存酸素量などといった指標の適切な環境制御が望まれる．これらの指標を観測するため，養液中

多種類のセンサを取り付けるのが一般的である．最適な培養液条件を見つけ，さらにその条件を満たすように養液の状態を制御する技術が必要となる．

空調技術

空気の温度や湿度，CO2 濃度は植物の呼吸や蒸散，光合成などの活動に影響を与えるため，結果として成長に大きく関わる．環境センシングの手法として，空気中の温度，湿度，CO2 濃度を測定するセンサが取り付けられることが多い．制御器としては，温度制御のためのヒートポンプ，CO2 濃度制御のために外気もしくは CO2 を混入させるためのエアポンプが取り付けられる．適切な空調環境を実現するための制御技術は複雑であり，今もなお多くの研究がされている [23, 24, 25, 26]．

照明技術

植物の主な成長機構は光合成であり，また全体の消費電力量のうち照明の占める割合は圧倒的に大きいため，高度な照明の制御技術が求められる．光合成を行う際にはクロロフィル（葉緑素）という色素が機能し，特定の波長を吸収する．吸収した光のエネルギーを用いて化学反応を誘起し，炭水化物を生成することで植物体を大きくする．クロロフィルは赤色と青色の二つの吸収ピークがあり，これらの波長が光合成に特に有効である．偶然にも赤色 LED と青色 LED の波長は共に吸収ピークに近いため，近年では LED が照明として用いられている．LED の照射時間及び照射強度は植物の成長に大きく影響するため，これらの変数を調整することによって生産性が変化する．光が強すぎると植物は全てのエネルギーを光合成には使用できず，余ったエネルギーを熱として植物体から放射されてしまい，LED のエネルギーは無駄になってしまう．このため，照明コストを考慮すると光を強くすればよいというわけではなく，コストと生産性のトレードオフを考慮した制御が重要になってくる．

3.2.3　Personal Food Computer

MIT Media Lab の Open Agriculture Intitiative (OpenAg) というチームが，オープンソースとして作成可能な完全制御型植物工場というコンセプトで Personal Food Computer (PFC) を作成し，その設計を公にしている [27]．PFC を組み立てるために必要な部品を網羅した BOM リストとそれらの加工法や設計図，そして PFC を稼働させるためのソフトウェアを GitHub 上に公開した．PFC の大きな意義は植物工場というハードウェアを統一したことによりレシピを共有できるようにしたことにある．一度 PFC を用いて植物を生育できるレシピを発見できれば，そのレシピは PFC を所有している世界中の人々に共有され，同様に植物を生育できる．

現時点では残念ながら，照明と冷却モジュールに関してオープンソースとして入手できるものが示されておらず，ソフトウェアも安定して動作するには至っていない．しかし，その不完全な情報ながらも世界中の多くの人が独自に PFC を作り上げる努力を行い，コミュニティも盛り上がっている．我々もまた公開された情報をもとに PFC を作成した．図 3.1 にその外観を示す．以下では，この PFC の詳細について説明する．

図 3.1: Personal Food Computer の外観

3.3　Personal Food Computer のハードウェア

3.3.1　システム構成

PFC は植物工場としての役割を果たすための多くの機能を備え，かつオープンソースとして作成可能であることを目標として設計された．部品の調達及び PFC の作成にあたり詳細な情報が GitHub 上に公開されている．しかし，世界中の人が作れることを目標に設計された簡易型の植物工場であるにも関わらず，一部の部品の入手が不可能であること，全ての設計図がインチ規格で書かれていること，作成には多くの加工機・ツールを要することなどから多くの人が簡単に PFC を作成できるとは言い難い．PFC を組み立てるためのキットは販売されておらず，全ての部品を一から加工する必要があるため作成には多くの時間を要する．ただ，農業や植物に対する専門的な知識を持ち合わせていなければ，簡易型の植物工場を実装するにあたってどのようなセンサやアクチュエータを備えればよいかは見当すらつかず，各センサ・アクチュエータの購入部品や実装方法をオープンソースとして公開する意義は大きい．PFC には 12 種類のアクチュエータと 8 種類のセンサが備わっている．それぞれのアクチュエータとセンサについて表 3.1 に示す．アクチュエータの制御及びセンサの読み取りを行うためにマイコンモジュールとして Arduino が使用されているが，Arduino 自身は USB で接続されている Raspberry Pi の命令を受取り，その命令に従って動作する．ハードウェアの構成要素の各々の役割から，A, E, S の 3 つの Type に分類した．A は環境に影響を与えるアクチュエータ，E は植物体を取り巻く栽培環境情

表 3.1: PFC を構成するセンサ及びアクチュエータ

Part Name	Function	Type	Target
Chamber Fan	Make air condition uniform	A	Air Circulation
Heater	Increase temperature	A	Temperature
Chiller	Decrease temperature	A	Temperature
Air Flush	Air Ventilation	A	CO2, Humidity
LED Light	Emit light (red and blue)	A	Light
Water Aeration Pump	Increase dissolved oxygen concentration	A	EC
Water Circulation Pump	Make water condition uniform	A	Water Circulation
Liquid Pump 1	Increase pH	A	pH
Liquid Pump 2	Decrease pH	A	pH
Liquid Pump 3	Add nutrient A	A	EC
Liquid Pump 4	Add nutrient B	A	EC
Liquid Pump 5	Increase water	A	Water Level
Air Temperature Sensor	Measure air temperature	E	Air Temperature
Humidity Sensor	Measure Humidity	E	Humidity
CO2 Sensor	Measure CO2	E	CO2
Light Intensity Sensor	Measure Red Light Intensity	E	Light
EC Sensor	Measure EC	E	EC
pH Sensor	Measure pH	E	pH
Water Temperature Sensor	Measure water temperature	E	Water Temperature
Water Level Sensor	Measure water level	E	Water Level
RGB Camera 1	Image top view	S	Leaf Area Index
RGB Camera 2	Image side view	S	Stalk Diameter

報を抽出するセンサ，S は植物体の情報を抽出するセンサをそれぞれ意味する．植物工場においてデータ収集基盤を実装する場合，ハードウェアの構成要素はこのいずれかに分類できると考えられる．

3.3.2　作成方法

組み立てるために必要な情報は以下の OpenAg の GitHub リポジトリに公開されている．このリポジトリには部品購入，部品加工，ワイヤ作成，配線，組立方法に関する情報がそれぞれまとめられている．以下に本リポジトリ内のファイルがどのような情報を示しているかを述べる．

`https://github.com/OpenAgInitiative/openag_pfc2`

購入部品

部品購入に関する詳細な情報はBOM/BOM_MASTERおよびBOM/BOM_Sub_Assemblysに記載されている．BOM_MASTER は PFC を組み立てるにあたり購入するべき全ての部品を網羅的に記載しており，BOM_Sub_Assemblys は後述する各 Sub Module 毎に必要な部品を記載している．しかし，一部の部品は配送の制約のために日本からでは購入できない，もしくは小売としての販売を行っておらず OpenAg のみが入手している商品がある．我々は日本で購入できない商品に関しては，代替となる購入先を発見し，代替購入先に関する情報を既存の BOM_MASTER に追加した．そして以下の URL で更新した BOM_MASTER を公開している．また，全ての部品を購入したときの合計金額は 30 万円以上となった．

`https://docs.google.com/spreadsheets/`
`d/1GCFbjkjvupg_Z1q32oTHkgxX73pUD-6EIp8D0dZehOw/edit#gid=713247668`

部品加工

PFC の筐体を構成する多くの部品は作成者自身が加工する必要がある．切断や穴あけ，曲げなどの加工をする必要がある部品は BOM_MASTER のシート “Raw Materials” としてまとめて記載されている．Raw Materials にはアクリル板やアルミ板，アルミフレームなどがあり，加工機としてレーザーカッター，CNC 切削機，ボール盤を用いて加工することが推奨されている．（これらの加工機を全て使いこなして加工することも一苦労である．）それぞれの Raw Materials をどのような形に加工すればよいかは全てディレクトリ CAD/GrabCAD/ 内に置かれているファイルを参照すれば良い．アクリル加工であればレーザカッターによる 2 次元加工なので，DXF 形式のファイルとして CAD/GrabCAD/DXF/ 内のファイルに記述され，曲げを含むアルミ板の加工などの 3 次元設計図は CAD/GrabCAD/Drawings/ に SLDDRW 形式で置かれている．このようにそれぞれの設計図は公開されているものの，PFC 作成のためには骨格となるフレームや電源部を支えるアルミフレーム，壁としての役割を持つアクリルパネルをそれぞれ材料から加工する必要があり，一から作成するには多くの時間を要するであろう．また，アクリルパネルは一片が 800 mm 程度のものもあり，加工が難しい．例えば，DMM.make で利用可能なレーザーカッターで加工可能なサイズの上限は 750mm 程度であり，作成できない．我々は PFC の作成にあたり一部のアクリルパネルはアクリル加工の専門業者であるアクリルドットコムに外注した．オープンソースとして設計図を公開し普及を目指すのであれば，DIY で作成可能なレベルであるべきだと思われるが，必要となる加工機のスペックが高く，DIY レベルでの作成可能とは言い難い．

ワイヤハーネスの作成

PFC を作成するにあたり，電力や信号を伝えるワイヤハーネスを自前で作成する必要がある．ワイヤー，コンタクト，ハウジングコネクタをそれぞれ購入し，かしめ工具を用いて作成する．作成するべきワイヤハーネスは Wire_Harness に全て記載されている．ここには，用いるワイヤーの種類，用いるワイヤーの長さ，ワイヤーの端のストリップする

長さ，用いるコンタクトの種類，用いるハウジングコネクタの種類，ワイヤーハーネスの ID がテーブル形式で記載されている．合計で 49 個のワイヤーハーネスを作成することになる．

配線

作成されたワイヤーハーネスをどのように配線するかは Wire_Reference に記載されている．各センサやポンプ，Raspberry Pi や Arduino などのマイコン，そして電源とどのワイヤーハーネスを用いて接続するかが記載されている．後述する Brain Module は各センサ・アクチュエータとの配線場所としての役割も持ち，接続場所の表現として印加電圧と Bus，Terminal の 3 つで一意の接続場所を表している．図 3.2 にそれぞれがどのように割り当てられているかを示す．ここに記載されている情報をもとに配線を行う．

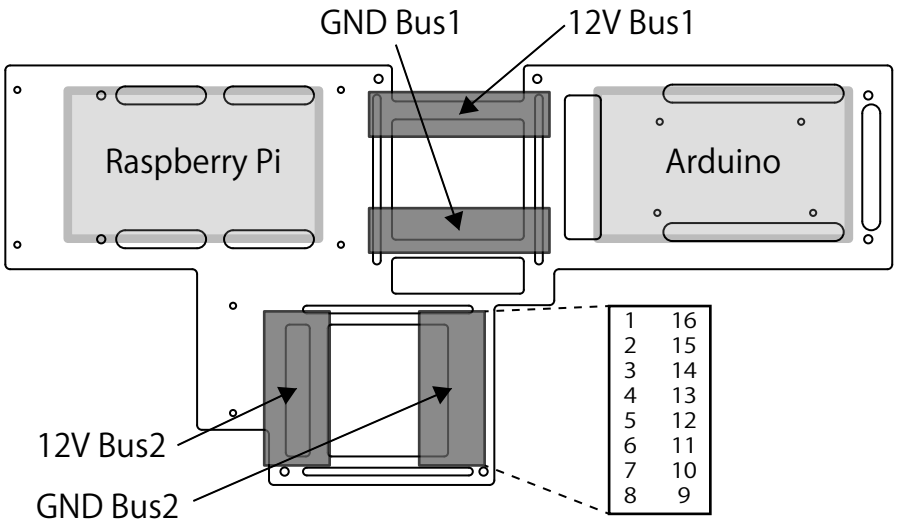

図 3.2: Brain Module における印加電圧と Bus，Terminal による接続場所の位置指定．

組立

PFC は表 3.1 に示す様々なセンサ及びアクチュエータによって構成される簡易型の植物工場である．PFC の全体の組み立てにあたっては 5 つの Sub Module である Frame, Electronics Panel, Brain Module, Light Panel, Water Manifole, Chiller から構成され，それぞれを独立に作り上げた後に，最後に全てを組み合わせるという手順を踏むと作成しやすい．Instructions の直下にそれぞれの Sub Module 名でのマークダウン形式のファイルが置かれ，各組み立て手順が記載されている．次の節 3.3.3 では，各 Sub Module について作成にあたっての注意点や改良した点などの GitHub 上には公開されていない箇所を主に説明していく．

3.3.3　Sub Modules

Frame

Frame は PFC の骨格のことであり，Raw Materials として購入した $0.75'' \times 0.75'' \times 8'$ のフレームより切削や穴あけを行って作成する．OpenAg が公開している図面はインチ規格で記述されているが，日本に存在する多くの工作機械はミリメートル規格であるため，換算する必要がある．改良した設計図の一例を図 3.3 に示す．

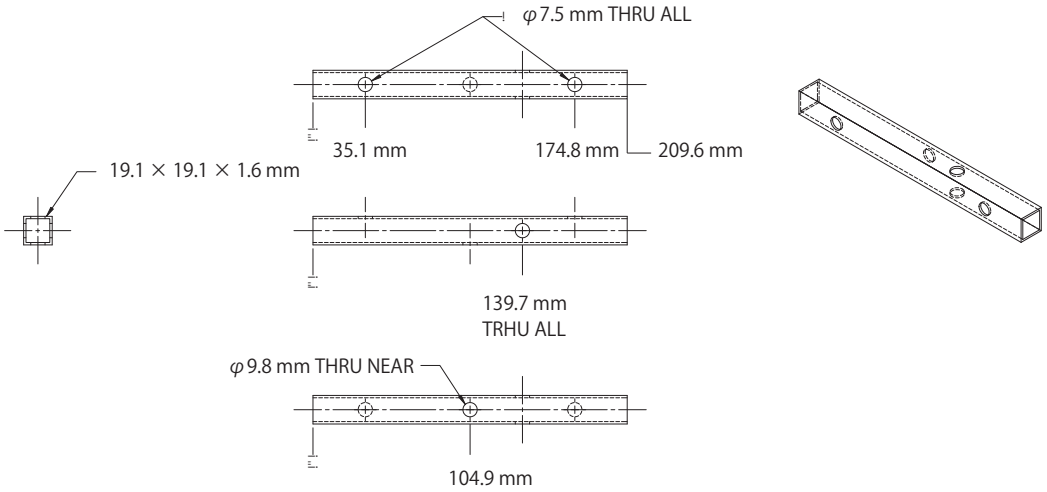

図 3.3: インチ規格から改良したフレームの設計図の一例．

Brain Module

Brain Module は PFC の各センサ・アクチュエータを制御し，操作するためのモジュールである．センサ値の取得と，アクチュエータへの動作制御は Arduino の AD コンバータや I2C 通信，PWM 出力をインターフェースとして行われる．Raspberry Pi はウェブサーバやデータベースとしての機能を持つ．また，Arduino は Raspberry Pi とシリアル通信を行なっており，Raspberry Pi から送られるデータをもとにアクチュエータを動作させる．Brain Module と各センサ及びアクチュエータの制御機構を表したシステム構成を図 3.4 に示す．

Light Panel

Light Panel はチャンバーに設置される LED モジュールを含めたものである．植物は主に光合成によって成長するため，どの LED モジュールを使用するかは生育に大きな影響を与え，植物工場において非常に重要である．しかし，OpenAg が公開している LED モジュールは現時点で小売りとしては販売されておらず，入手できない．我々は代替となる LED モジュールとして，lampwin より販売されている交流 100 V，45 W で駆動する LED モジュールを購入し，実装した [28]．また，使用する LED モジュールを変更したことによ

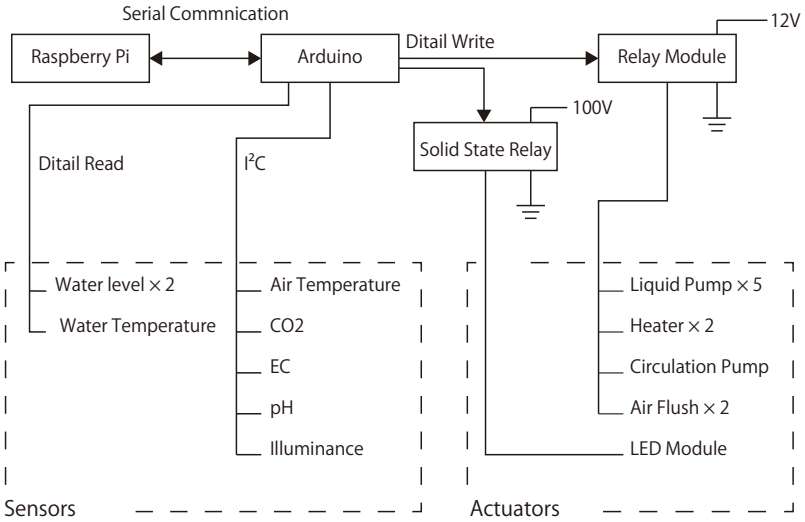

図 3.4: Brain Module とその周辺のシステム構成図

り，PFC のチャンバーにおける取り付け方法も大きく変化した．既存の PFC の設計では，光量は到達距離の二乗に反比例するため，800 mm 程度の高さがある PFC において代替の LED モジュールを天井付近に設置した場合，植物体が生育するに十分な光量を得ることが難しかった．そこで，図 3 に示したように LED モジュールを植物体から高さ 300 mm 程度に設置するようにした．

Water Manifold

Water Manifold は養液のセンシング及び制御を行うモジュールである．備え付けられるセンサとして，水温センサ，水位センサ（2 つ），EC センサ，pH センサがある．アクチュエータとしては，養液の酸素濃度を高めるためのエアポンプ，養液をトレー内で循環させるためのサーキュレーションが取り付けられている．また，Electronics Panel に取り付けられた pH アップ調整剤，pH ダウン調整剤，液肥 A，液肥 B，水を供給する 5 種類のポンプチューブが設置される．このモジュールにおいて，それぞれがどの位置に取り付けられるかを図 3.5 に図面と共に示す．センサやアクチュエータの設置場所は読み取られるセンサ値や養液内環境に影響を与えるため，全ての実験を通して統一する必要がある．

また，OpenAg の公開情報に従って Water Manifold の設置を行うと生じる不具合が幾つか存在するため，以下に問題点とその対策を列挙する．

- pH アップ調整剤や液肥 B などの粘性（表面張力係数）が小さい液体は，ポンプが稼働していなくても液垂れしてしまい，意図せずに養液環境を変化させてしまうことがある．対策として液垂れ防止ツールの使用や液体チューブの径を小さくすること，チューブの先を垂直下方向に向けないで設置することなどが挙げられる．

- 液肥 B がチューブ内において，凝固して結晶となり流路を塞いでしまうことがしばしば生じる．流路が塞がれたときはチューブを新しいものに交換する必要がある．
- 養液の底にエアストーンを設置すると，エアストーンとチューブの隙間から養液が侵入し，空気の流路を塞ぐためにエアポンプが稼働しなくなる．シーラントなどでチューブエアストーンの隙間を塗り，下に台を置いて底上げすることでエアストーンの流入口を水位より高くする必要がある．
- Water Manifold は透明のアクリルを使用しているが，光りを通すために養液中に大量の藻が発生してしまう．藻は養液中の栄養を吸収するために，藻が大量発生すると植物の生育に悪い影響を与えてしまう．アクリルを不透明な色に変更するか，不透明なシートで覆うかなどで遮光する必要がある．

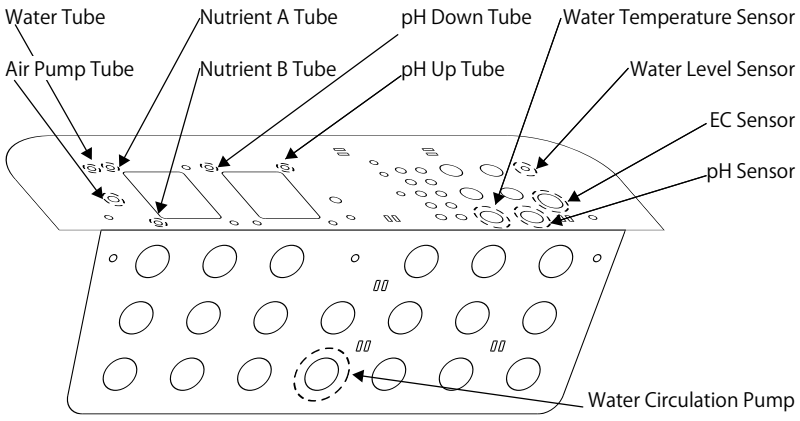

図 3.5: Water Manifold における，各センサ，アクチュエータ，チューブの取り付け位置

Electronics Panel

Electronics Panel は植物を育てるチャンバーの側面に設置されるアクリルパネル及びそこに取り付けられたセンサやアクチュエータ，マイコンなどのモジュールを合わせた総称である．センサやアクチュエータなどの設置場所を記した Electronics Panel の図面を図 3.6 に示す．

Chiller

Chiller はチャンバー内の気温を下げるためのモジュールである．一般的に，植物工場内の気温は植物の呼吸に大きな影響を与えるため，気温調整は重要である．しかし，OpenAg が公開している Chiller Module は小売りとしては入手不可能であるため，使用することが出来

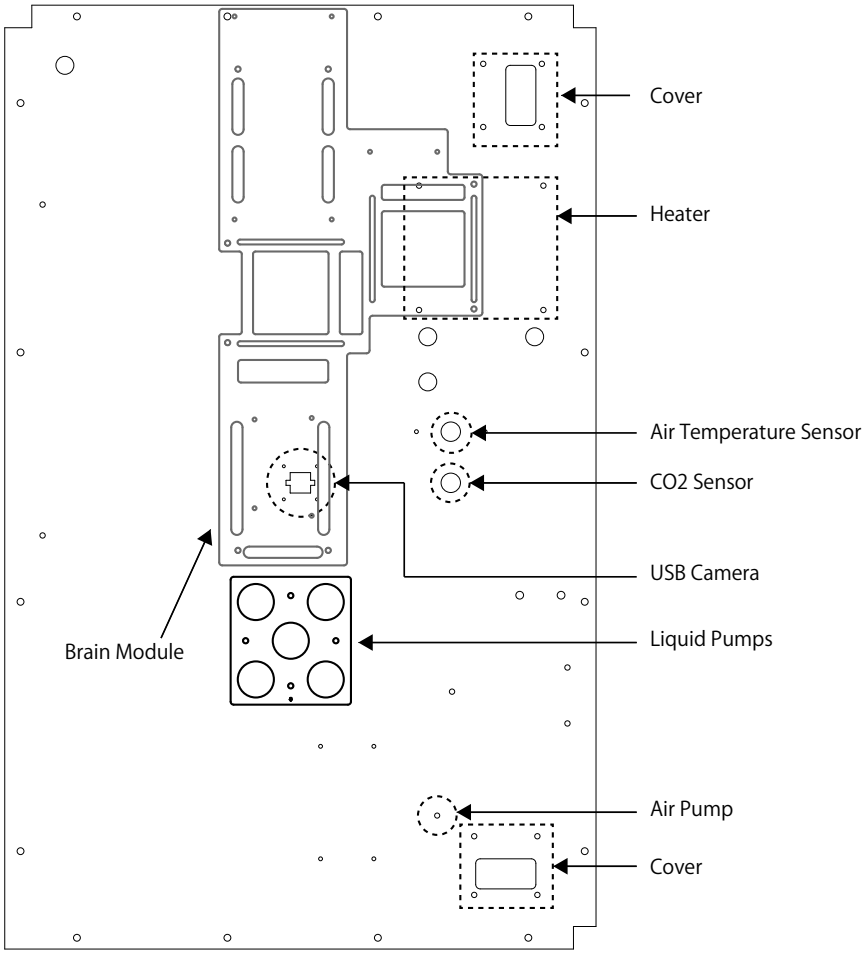

図 3.6: センサやアクチュエータなどの場所を示した Electronics Panel の図面.

ない．我々はこれの代替として，RIGID の DV1920E-P (24V) を購入した [29]．DV1920E-P は，OpenAg が使用している Chiller Module と同様に 24 V 駆動であり，配線などは同じようにして組み込むことが可能となる．この代替モジュールが実際に動作して，Arduino で制御可能であることを確認した．

3.4　Personal Food Computer のソフトウェア

3.4.1　ソフトウェアパッケージ openag_brain

OpenAg は openag_brain というソフトウェアパッケージを公開している．Raspberry Pi (Raspi) にインストールすれば，PFC の全てのアクチュエータとセンサを制御できる Robot Operating System (ROS) で記述されたプログラムが実行可能になる．OpenAg は植物工場の制御を “Act”, “Plan”, “Sense” の 3 つの行動のサイクルとして考え，ROS を採用した．ROS は非同期的に動作する複数のノードから構成されるプログラムを実行し，

特定の制御を担当しているノードにエラーが起きてもプログラム全体が停止することはなく，ロバスト性がある．

また，openag_brain がインストールされた Raspi には Arduino を通じてハードウェアを制御する以外にもウェブインターフェースやデータベースとしての機能を持つ．データベースのフレームワークとしては CouchDB[30] が使用されている．以下では，openag_brain やウェブインターフェース，データベースの構造について詳細を述べる．

3.4.2　ROS

ROS Nodes

openag_brain は複数の ROS ノードから構成され，それぞれが異なるプロセス上で動作している．ノードの役割は，Arduino や USB カメラなどの外部モジュールとのインターフェースの役割を行うノード，データベースを操作するノード，制御値を計算するノードなど多岐にわたる．Raspi 上で動作する ROS ノードのシステム概要を図 3.7 に示す．

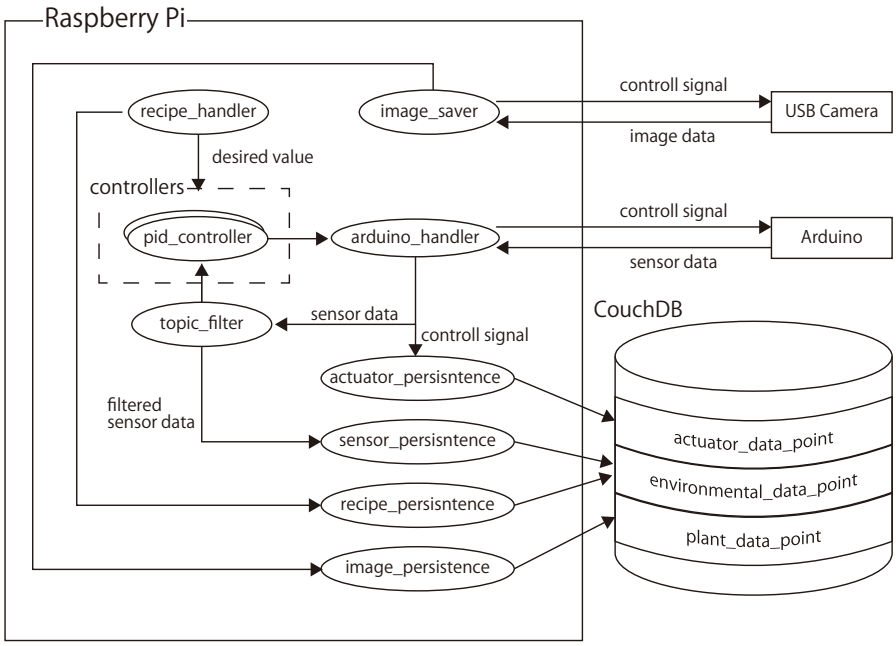

図 3.7: Raspberry Pi 上で動作する ROS ノードのシステム関係図．

launch ファイル

ROS フレームワークでは複数の ROS ノードを実行する場合，各 ROS ノードの実行方法をまとめて記述した launch ファイルを用いることが推奨されている．我々は実験を行うにあたり，主に以下二つの launch ファイルを用いている．

personal_food_computer_v2.launch　PFC で栽培実験を行う際に実行するファイル．

setup_solution.launch 栽培実験前に一定の養液を作り出すための実行ファイル.

実行するときは以下のコマンドを実行する

```
$ rosrun openag_brain main personal_food_computer_v2.launch
```

3.4.3 Firmware

Arduino に書き込まれるスクリプトはディレクトリ openag_brain/firmware 内に置かれている．書き込みのツールとして platformIO [31] を使用している．platformIO は同一のスクリプトで様々なマイコンに書き込めることを可能にしたオープンソースのプラットフォームであり，openag_brain でもファームウェアの書き込みに使用されている．以下のコマンドを実行することでスクリプト，openag_brain/firmware/src/src.ino が書き込まれる.

```
$ rosrun openag_brain firmware -t upload
```

Arduino で動作するプログラムは Raspi とのシリアル通信を通して，各アクチュエータの制御と各センサの値の読み取りを行う必要がある．図 3.8 に示す．openag_brain/firmware /lib/openag_firmware_module/openag_module.h にて抽象クラス Module を定義し，各センサ・アクチュエータはこれを継承した作られるクラスで初期化されるインスタンスによってその動作が記述されている．抽象クラス Module は抽象メソッド update() を宣言している．各インスタンスにおいて update() メソッドが呼び出されると，センサにおいてはインスタンスが保持するセンサ値が更新され，アクチュエータにおいてはデジタル出力ピンの出力電圧を更新する.

3.4.4 Web Interface

OpenAg の Github には “openag_ui” というレポジトリがあり，ここに記された順に従ってインストールを行えばブラウザを通して PFC の操作やセンサ値の時間遷移を見ることができる．ブラウザ上に表示されるセンサ値の時間遷移のグラフの一例を図 3.9 に示す.

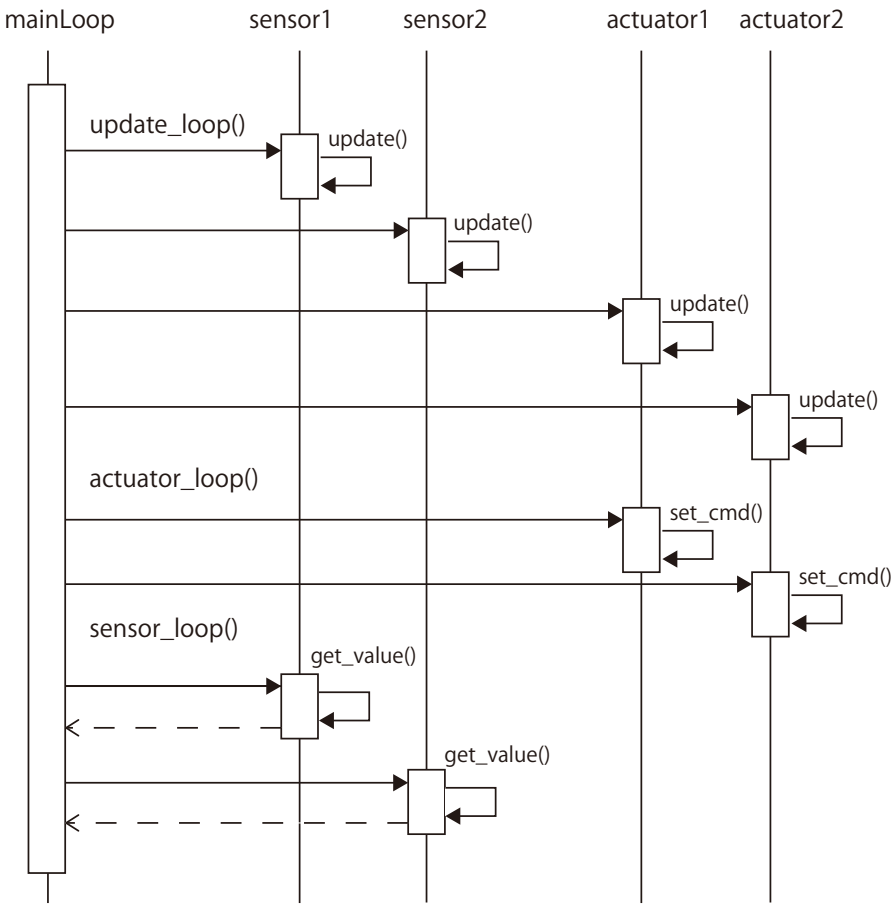

図 3.8: Arduino 上で動作するプログラムのシーケンス図.

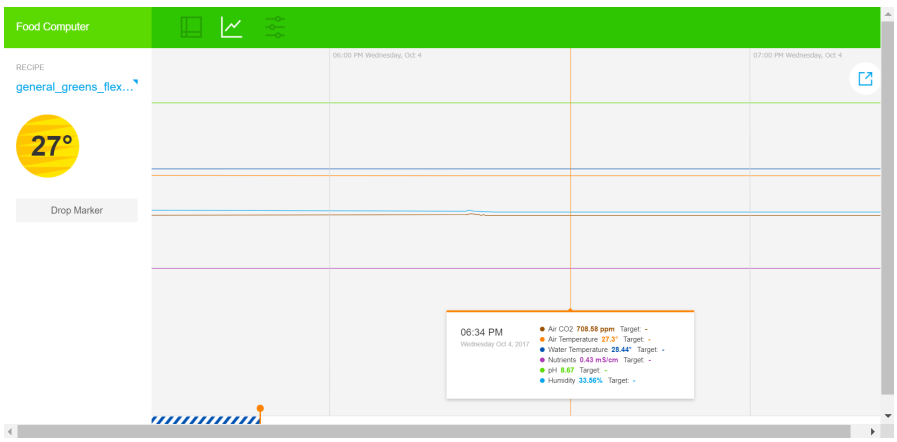

図 3.9: openag_ui をインストールした後に表示される PFC のウェブインターフェースの画面例.

3.4.5　レシピのフォーマット

レシピは PFC のチャンバー内の各センサの目標値や各アクチュエータの動作を決定する．PFC においてレシピは json 形式で記述される．レシピの一例をソースコード 3.1 に示す．レシピの環境制御に関する情報はプロパティ phases で定義される．step が 1 周期内での各時刻における環境制御値を示す．例えばソースコード 3.1 では，レシピを開始した時刻を 0 として最初の 17 時間は気温は 23 °C になるように制御される．プロパティ format には flex_format がよく用いられ，step がどのように記述されるかを定義する．ここで定義された値とセンサ値を基にアクチュエータを制御する．

ソースコード 3.1: レシピの一例．JSON フォーマットで記述される

```
{
  "_id": "pfc_stable_grow",
  "format": "flexformat",
  "version": "1.0",
  "seeds": ["lettuce"],
  "plant_type": ["lettuce"],
  "certified_by": ["Tatsuya Iizuka"],
  "author": "Tatsuya Iizuka",
  "date_created": "2017-12-7",
  "phases": [
    { "name": "early",
      "cycles": 30,
      "time_units" : "hours",
      "step":
        { "air_temperature": [
            {"start_time": 0, "end_time": 17, "value": 23},
            {"start_time": 17, "end_time": 24, "value": 21}],
          "light_intensity_white": [
            {"start_time": 0, "end_time": 17, "value": 1},
            {"start_time": 17, "end_time": 24, "value": 0}],
          "air_flush": [
            {"start_time": 0, "end_time": 0.25, "value":1},
            {"start_time": 8, "end_time": 8.25, "value":1},
            {"start_time": 16, "end_time": 16.25, "value":1}],
          "water_potential_hydrogen": [
            {"start_time": 0, "end_time": 24, "value":6}]
      }
    }
  ]
}
```

表 3.2: 作成されたテーブルの種類とその役割.

テーブル名	役割
environmental_data_point	各環境のセンサ値およびレシピで定義された目標値の保存
plant_data_point	植物の画像データの保存
acutator_data_point	アクチュエータの動作に関する情報の保存

表 3.3: environmenta_data_point に保存されるドキュメントが持つ各フィールドに関する説明.

フィールド名	値の意味	データ型
environment	どの PFC によって栽培されたかを表す	String
is_desired	レシピが定義する目標値か，センサ値であるかを示す	Bool
timestamp	このドキュメントが保存された時刻を示す	Float
value	指定された環境変数の値	Float
variable	どの環境変数を示しているか	String

3.4.6　CouchDB

CouchDB はオープンソースのデータベースであり，JSON 形式のドキュメントとして各テーブルに保存される．HTTP を通じて操作可能な API が多く用意されており，ブラウザからでもデータベースの閲覧や操作が可能となる．PFC において各センサ値やアクチュエータの動作履歴などを保存する役割を持つ．openag_brain をインストーすると CouchDB は Raspberry Pi 上にインストールされるが，我々は LAN 内に仮想マシンを用意し，全てのデータをその仮想マシン (VM) 上に保存するように設定を変更した．VM 上に作成されているテーブル名とその役割を表 3.2 に示す.

また，environmenta_data_point に保存されるドキュメントのフィールドには全てのドキュメントに付与される “_id”，“_rev” 以外に 5 つのフィールドをもっており，そのフィールド名と意味，データ型について表 3.3 に示す.

CouchDB はスキームレスなデータベースであるため，ドキュメントはここに説明された以外のフィールドを持ってもよい．しかし，表 3.3 に示した 5 つのフィールドで必要十分な情報となるため，センサ情報を格納するための全てのドキュメントはこの形式で保存されることになっている.

また openag_brain のデフォルトには実装されていないが，画像を VM 上に保存するように我々は改良した．VM 上のディレクトリ /home/iizuka/ImageDatabase 内に PFC で取得された全ての画像を保存している．ファイル名は撮影時刻の UNIXTIME（JST) とし，拡張子は JPG である.

3.4.7　存在するバグ

ソフトウェアパッケージ openag_brain は公開情報の指示通りにインストールを行って稼働させても，正しく動作せずにエラーを起こして停止したり，誤動作したりすることがある．以下に，生じうるバグについて列挙した.

- 稼働してからある程度の時間 (数時間から数日) を置くと Arduino の I2C センサ読み取りにエラーが生じ，以降全てのセンサの読み取りが出来なくなることがある．Arduino に書き込まれるプログラムを変更してセンサの読み取り頻度を下げることで，このエラーは生じなくなった．

- レシピで目標値を定義していない時間帯においては目標値が 0 として出力されてしまう．例えば pH 制御目標値を設定する場合，レシピの回転期間中は常に明示的に値を指定していない時間帯では目標値が 0 として設定されてしまい，pH ダウン調整剤を駆動し続けることになってしまう．

- インストール直後に CouchDB のサーバが Raspi 上にたてられるが，初期設定では外部からのアクセスを全て許してしまう．Raspi にグローバル IP が割り当てられている場合，外部の悪意のあるものによってデータベースの内容に変更を加えられたり，削除されたりする危険性が生じる．

- 液肥のポンプ駆動は firmware のクラス DoserPump によって生成されたインスタンスが制御している．このインスタンスは Raspi より液肥の投与速度 $\mathrm{ml\,hour^{-1}}$ が入力として与えられる．しかし，これはポンプの駆動がしたときの投与速度が $1000\,\mathrm{ml\,hour^{-1}}$ を仮定しているが，我々の環境では $8450\,\mathrm{ml\,hour^{-1}}$ として設定を変更し，firmware への書き込みをやり直した．

3.4.8　独自の追加実装部分

本章におけるこれまでの記述はシステムズなどを含め，我々が現在稼働させている Personal Food Computer に関する説明であり，OpenAg が設計したものに我々が改良した部分を加えて一緒くたに説明を行ってきた．本節ではこれまでの説明を振り返り，我々が追加した部分に関して明示する．

ハードウェアの独自実装

節 3.3.3 の Light Panel で述べたように OpenAg が指定する LED モジュールは入手不可能であったため，代替となる LED モジュールを選定し，植物体に十分な光量を与えられるように底面から 30 cm 上空に固定した．また代替 LED モジュールは AC 電源で動作するため，スイッチング機構は別途実装する必要があった．AC 電源のスイッチングを行えるソリッドステートリレーを用いて Arduino のデジタルアウトプット（5 V 出力）によって制御可能にした．また，植物体の天井からの画像を取得するために，カメラを設置する必要がある．植物体との距離が 300 mm と近い一方で植物体の存在する範囲は 400 mm×300 mm 程度であるため，通常のカメラでは画角に収まらない．そこで USB インターフェースをもつ画角が 120° と広角な BUFFALO 製のカメラ BSW200MBK を購入し，設置した [32]．

カメラによる画像取得にあたり，代替 LED モジュールは交流駆動のため 50Hz で点滅している．CMOS カメラに生じるローリングシャッター現象により，50Hz で点滅を繰り返す照明下では取得画像に明暗の縞模様が生じてしまう．撮影時にパラメータの一つである F 値を調整することで，縞模様が覗かれた画像を取得することに成功した．

ソフトウェアの独自実装

ソフトウェアパッケージ openag_brain にはそもそもカメラを操作して画像を定期的に取得するための実装が施されていなかった．そのため，我々は画像を定期的に取得する “image_saver.py” 及び，取得画像を VM 上のデータベースに保存する “image_persistence.py” を実装した．Raspi から VM に画像を送信する際には SCP を用いた．

またデータベース上のテーブルとして，アクチュエータがいつ・どのような動作をしたかのログを保存する “actuator_data_point” を作成し，そこに各ログを保存する ROS ノード “acutator_persistence.py” を実装した．

他にも，実行するレシピの内容を VM 上から動的に変更可能なように “recipe_update.py” を記述し，レシピの仕様を一部改良した．詳細は 4.4 で述べる．

3.5　データ収集における栽培実験環境

3.5.1　栽培品種

PFC を用いた栽培実験において，栽培対象とする品種を 1 種類に限定した．対象とした品種はサカタのタネより販売されているベビーリーフレタス・スベンである [33]．気温は 15~20 °C でよく育ち，葉長が 15 cm 以上になったときが収穫の目安となる．

3.5.2　準備

葉長が数 cm 以上の栄養生長期にある植物は，植物体が大きくなるほど成長速度も大きくなる．我々は環境応答として成長量を主に観測したかったため，植物が一定以上の大きさになってから PFC のチャンバーに移植した．移植する前の苗を育てる環境を PFC とは別に用意し，なるべく一定の条件を保つようにして生育を行った．設備の外観を図 3.10 に示す．環境制御に関して，気温の制御は出来ないが，照明は PFC で用いられている LED モジュールと同じものを毎日 22 時間照射し，トレー 2，3，4 においては養液の液肥 A 及び液肥 B の濃度が 0.1%となるようにした．また，溶存酸素濃度を高めるためにエアポンプも設置している．この設備における植物は，播種からの日数に応じて 4 つのエリアに区分して育てられている．最初に水耕栽培用のスポンジを 10 個用意し，スポンジ 1 個あたり種を 2 つ撒いてトレー 1 で育てる．播種した日を 1 日目と数えたとき 1~7 日目は水道水のみを供給し，発芽させる．トレー 1 で育った株のうち生育の早い 6 株を選定し，トレー 2 に移植した後，養液で満たす．水耕栽培において養液に多くの栄養が含まれる場合，一般的に藻が大量に発生しやすい．養液に光りが当たらなければ藻の発生量は抑えられるため，トレー 2 は養液に外部から光が極力当たらないようにカバーなどで遮光している．このトレー 2 のなかで 8~14 日目になるまで育てる．播種後 15 日を経過したものはトレー 2 からトレー 3 に移植される．トレー 3 においては実際に PFC でそのまま栽培可能なように発泡スチロールに移植した状態で生育させる．トレー 2 で生育の早かった上位 4 株を選定し，トレー 3 に移植する．ここで，トレー 3 に養液を 10 L 注ぐ．トレー 3 に移植された植物は PFC に移植されるまでこの環境で最大 2 週間育てられる．

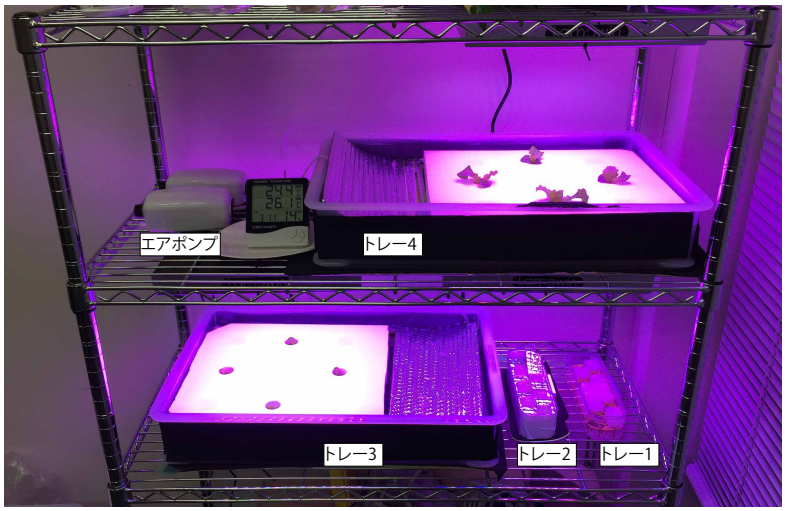

図 3.10: PFC へ移植するまでの株を育てる育苗環境の外観．種はトレー 1 に播かれ，7 日後にトレー 2，14 日後にトレー 3，21 日後にトレー 4 へと移植される．

3.5.3　移植

播種されてから 21 日以上経過した後に PFC に移植され栽培実験が開始する．トレー 3，4 で用いている発泡スチロールはそのまま PFC の養液トレーに設置できるサイズであるため，そのまま移植すれば良い．植物を置いた時点でレシピを開始すれば栽培実験が始まるが，始める前に PFC において準備することをいかに列挙する．

rosrun による setup_solution.launch の実行

レシピを回転する前に適切な養液を作り出すために，setup_solution.launch を実行する．これにより水位は高位の水位センサの高さまで水を供給する．その後，EC が 1.0 を超えるまで液肥 A と液肥 B を投与し，最適な pH になるように pH 調整剤を投与する．ここで，setup_soultion.launch を実行するにあたり，各液肥チューブが液肥で満たされていなければならない．さもなければ，片方の液肥のみが多く投与されてしまうことになる．

テープによる遮光

発泡スチロールとトレーの隙間や，Water Manifold の穴などから LED の照明が養液を照らすため，対処しないと大量の藻が発生してしまう．そのため，白いテープで隙間を塞ぎ社交する必要がある．発泡スチロールとトレーの隙間をテープで覆うことは，画像処理をしやすくなるという効果もある．

カメラから得られる画像における植物体の位置確認

最後に確認する必要があるのが，取得画像における植物の位置である．広角カメラ

を使用しているが，植物体との距離が小さいためにフレームアウトしやすい．PFC 内のトレーを設置した後に，レシピを回転させる前にカメラの画角に植物体がしっかりと収まっているかを確認する必要がある．

以上の注意事項を踏まえてレシピを回転させれば，栽培実験間で著しい環境の違いは生じないと考えられる．

第4章

レシピ探索のためのデータ解析基盤の実装

4.1　概要

植物工場の利益を向上させるためにレシピの改良は非常に重要である．しかし，未だにレシピ改良の方法論は確立されておらず，各企業の研究開発者が持つ経験と勘に頼った試行錯誤によって行われている．植物の生態生理学の知見を利用することである程度良いレシピを見つけることはできるが，制御するべき環境要素は照明や気温，CO2 濃度，EC，pH など多岐にわたり，このような多次元の要素が複雑に影響し合う条件下での植物の環境応答はブラックボックスであるため，全体的最適化は出来ない．熟練者の勘や生態生理学に基づく試行錯誤では，このような多次元の要素から構成されるレシピの最適化は難しい．そこで我々は，データに基づいたレシピ改良手法の確立を目的として研究を進めてきた．3 章で述べたデータ収集基盤を用いて，レシピ探索を行うためのソフトウェア基盤の実装を行ったため，本章で詳述する．続く 4.1.1 節，4.1.2 節では，レシピ探索とはなにか，そのためには何が必要であるかという一般的な方法論を述べ，4.2 節以降では 3 章で述べたデータ収集基盤 Personal Food Computer における具体的案実装の詳細を記した．

4.1.1　レシピ探索

本稿ではレシピ探索を，植物工場における植物の生育を早めるような環境の制御目標値を探索することと定義する．植物工場は，様々な制御器を備え，それらによって各環境値を所望の目標値に制御でき，それによって植物の生育が早めることができる．制御器を A，環境値を E，植物の状態を S としたときの，この植物工場における環境制御と植物成長の一連のプロセスは，図 4.1 に示すモデルによって表現できると考える．

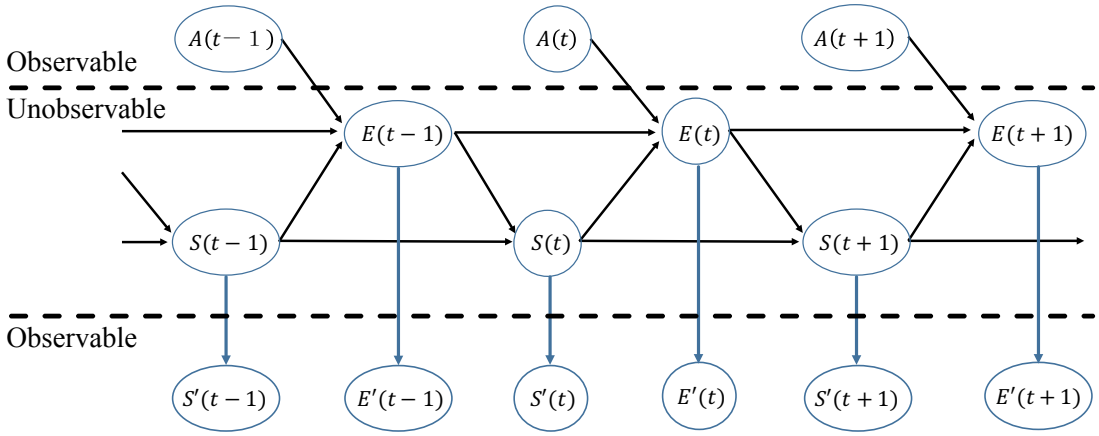

図 4.1: 植物工場における制御器 A，環境値 E，植物状態 S の時間遷移の関係性のモデル

このモデルでは，$A \to E, E \to S, S \to E$ の向きで影響をあたえることを仮定している．$A \to E$ の関係性は，制御器の操作量によって環境の変化量が決まることを表している．例えば，工場内の温度変化量は制御器であるヒータのオンオフ状態に依存するであろう．$E \to S$ の関係も植物の成長速度は環境に大きく影響することを表現している．そして，$S \to E$ の関係性の具体例として植物の呼吸や光合成が環境中の CO2 濃度を変化させ

ることが挙げられる．E と S を直接観測することは出来ないが，植物工場に環境及び植物のモニタリングシステムが備わっていれば，それぞれを部分観測した E' 及び S' として得られる．ここで部分観測とは，真の値に正規分布などの一定の隔離分布に従う誤差が加算された値としている．

植物工場の動作を含めた全体最適化を行う場合は A を含めた最適化問題として考えれば良い．しかし，A を含めた最適化問題として捉えると考慮するべき変数が増え，問題を単純化させるため，本稿における実験では所望の環境 E に制御可能であるという前提をおくことで，$A \to E$ の関係性を考慮する必要をなくした．

このようなモデルを仮定した上で，レシピを改良していくためには次の二つの処理が必要となる．

- センサ値に基づく $E \to S, S \to S,$ のモデル化
- 得られた S から生育の良し悪しの評価としてのスコア化

レシピを改良していくためには，得られた時系列のデータ E, S から $E \to S$ を予測可能なモデルを作る必要がある．そして，次に予測した S に対してその良し悪しを定量的に評価するためのスコア化が必要となる．これにより E からスコアを予測することが可能となり，スコアを最大化するような強化学習の適用や，バンディットアーム問題として解決可能な問題に帰着させることができる．

4.1.2　実装されるべきソフトウェア要件

植物工場で収集したデータをそのまま，4.1.1 節で示した方法論に適用することは難しく，通常は多くのソフトウェア実装を要する．主に必要となる実装要求は以下の三つである．

1. データベースに保存された生データから時系列の A, E, S を求めるデータ解析
2. A, E, S からレシピの良し悪しを表すスコアを計算するアルゴリズム
3. スコアを踏まえた上での効率的なレシピ探索手法を可能とするサンプリング手法

残る本章では，これらについて行った実装について詳述する．

4.2　生データから各系列データの抽出

4.2.1　データベースに保存された情報の抽出

3 章で述べたように PFC ではデータベースとして CouchDB を使用しており，用意された API を利用して所望の条件のデータを抽出する．保存されているデータは，観測・制御目標とされた各環境センサのデータ，レシピの稼働履歴に関するデータの 2 種類である．以下にそれぞれの取得されたデータとその抽出のための実装について記す．

環境データの抽出

CouchDB には Design Document と呼ばれる，データベースに投げるクエリを定義するドキュメントを保存することができる．クエリを表現する JavaScript のコードを文字列として渡すことで，特定のドキュメントのみを受け取ることができる．データベースの操作を行うクラス DBManager を実装した．このクラスのメソッド get_observed_data() を実行することで，特定の環境センサで取得されたデータの時系列データを取得することができる．また，このクラスはメソッド get_target_data() が実装されており，レシピで定義された各環境の目標値を時系列として返す．これら二つのメソッドを用いて，温度センサの取得値と目標値の時間遷移を図 4.2 に示す．たしかに，センサ値が目標値に追従して変化していることが分かる．

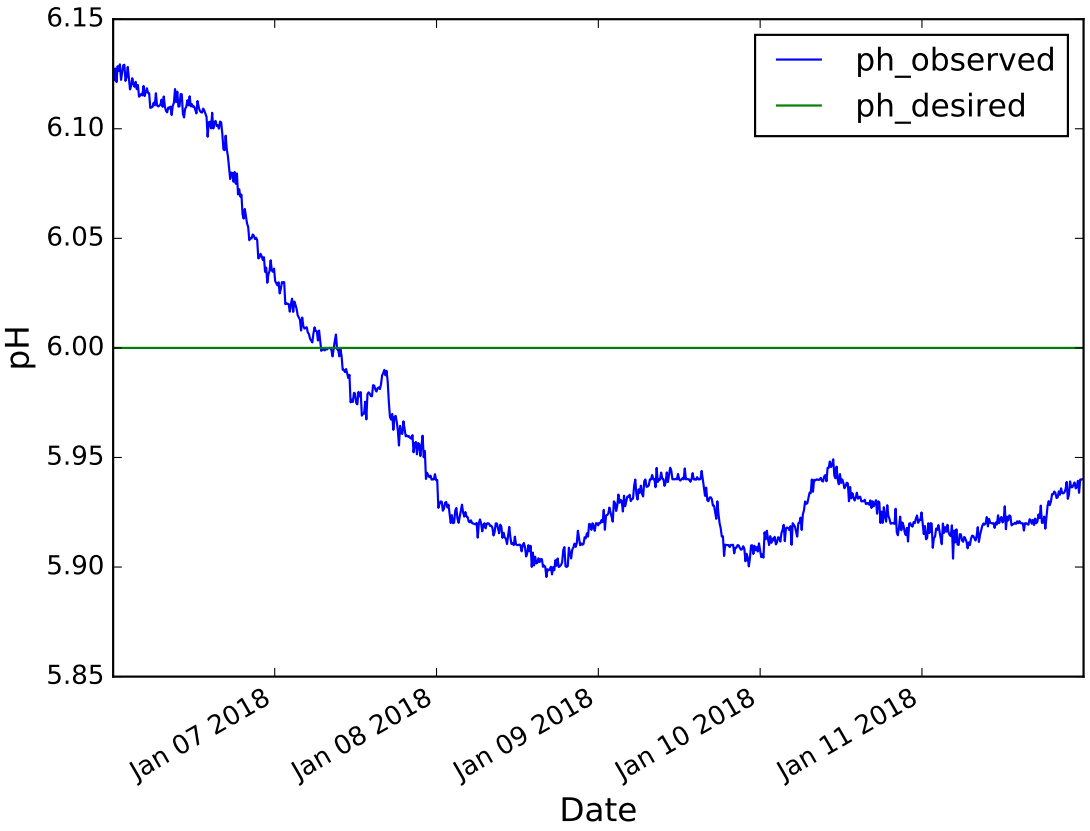

図 4.2: データベースに保存されていたセンサ値および制御目標値の時系列データ

レシピデータの抽出

レシピは開始される時に，3.3 で示されたレシピのフィールド “variable” が値 “recipe_start” として，終了する時には “recipe_end” としてデータベースに保存される．よって，これらのドキュメントを抽出すればよく，クラス DBManager のメソッド get_recipe_history() を実行することで，各レシピに対する開始時刻及び終了時刻がわかる．

4.2.2　植物状態に関するデータの抽出

植物体の状態に関するデータとして，天井と背面に実装された画像から得られる情報と，光合成速度などの環境センサの変化から得られる情報がある．以下にそれぞれについて述べる．

画像解析による葉面積抽出

天板に取り付けられたカメラにより，1 時間間隔で植物体の画像が取得される．株は白い発泡スチロールに設置されるため，葉の領域と背景の区分は難しくない．植物体の画像から葉面積や重心位置など形に関する情報を計算するクラス PlantImageProc を実装した．メソッド get_plant_data() を実行すれば，葉面積と重心位置に関する情報を返してくれる．葉の領域を正しく行えた画像の一例を図 4.3 に示す．葉の領域検出にあたり，色閾値の設定に関する譲歩などを含んだ幾つかのパラメータを要する．多くの画像において同じパラメータで処理が成功するが，失敗する画像も存在する．その場合はパラメータの調整が必要となる．

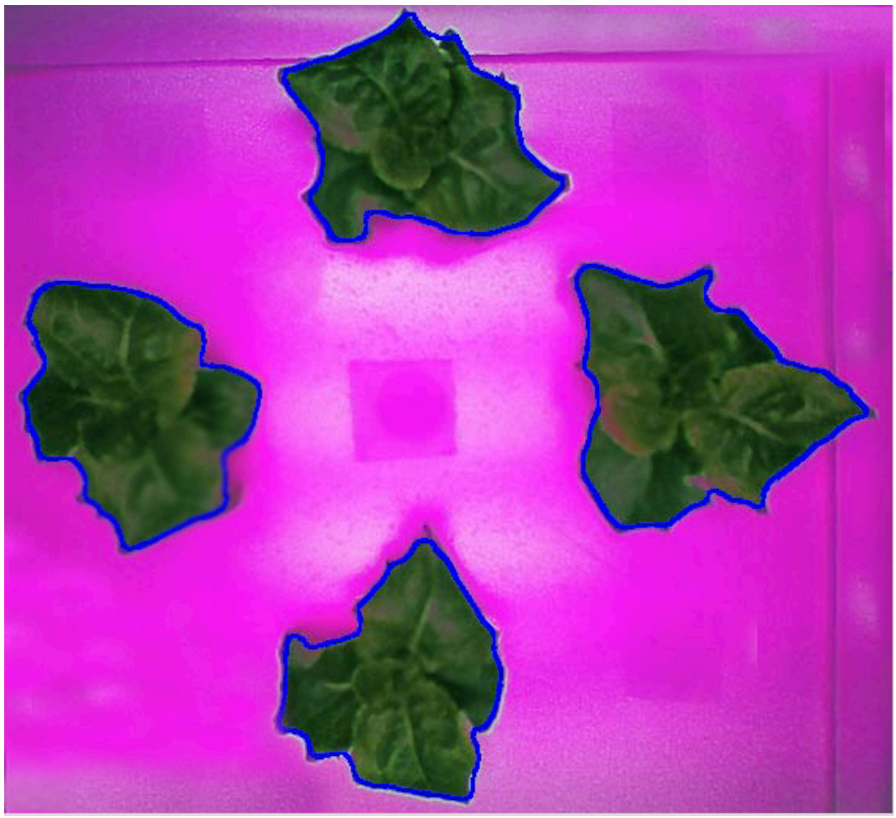

図 4.3: 天井から植物体を撮影した画像における葉の領域抽出．検出された葉の領域がプログラムにより青線で描画されている．

CO2 の変化量解析による光合成速度の抽出

グロースチャンバーにおける光合成速度は CO2 の減少速度を求めることで算出される．光合成速度は植物生育の指標としてよく用いられている．CO2 とその時間微分の時間遷移を図 4.4 に示す．図より CO2 濃度は変動が大きく，光合成速度も不規則に変動していることが分かる．データに一定の法則性を見出すことが難しかったため，本稿においては植物状態 S に関するデータとして光合成速度を用いなかった．

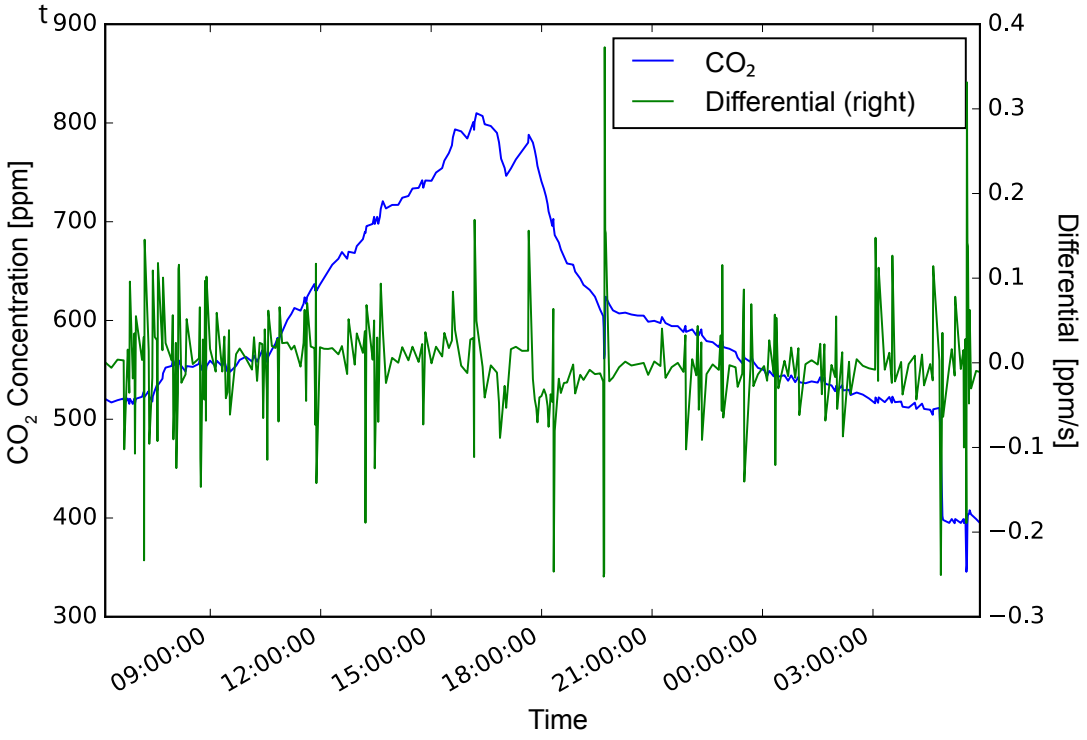

図 4.4: CO2 の時間変化とその時間微分による光合成速度の遷移図．

4.2.3　フィルタリングによる時系列葉面積の平滑化

植物は動くため，実際の葉面積が大きくなっていなくともカメラより観測された画像で見られる葉面積の値が変化することはよく起こる．そこで，得られた葉面積の値は，真の値に正規分布の誤差が乗ったものとして観測されるという仮定のもとで平滑化を行うことで，観測データからより有意義な情報を得ることができる．平滑化を行う関数としてクラス KalmanFilter を実装した．このクラスを用いて観測された葉面積値を平滑化した．観測値および平滑化後の値の時間遷移を図 4.2.3 に示す．

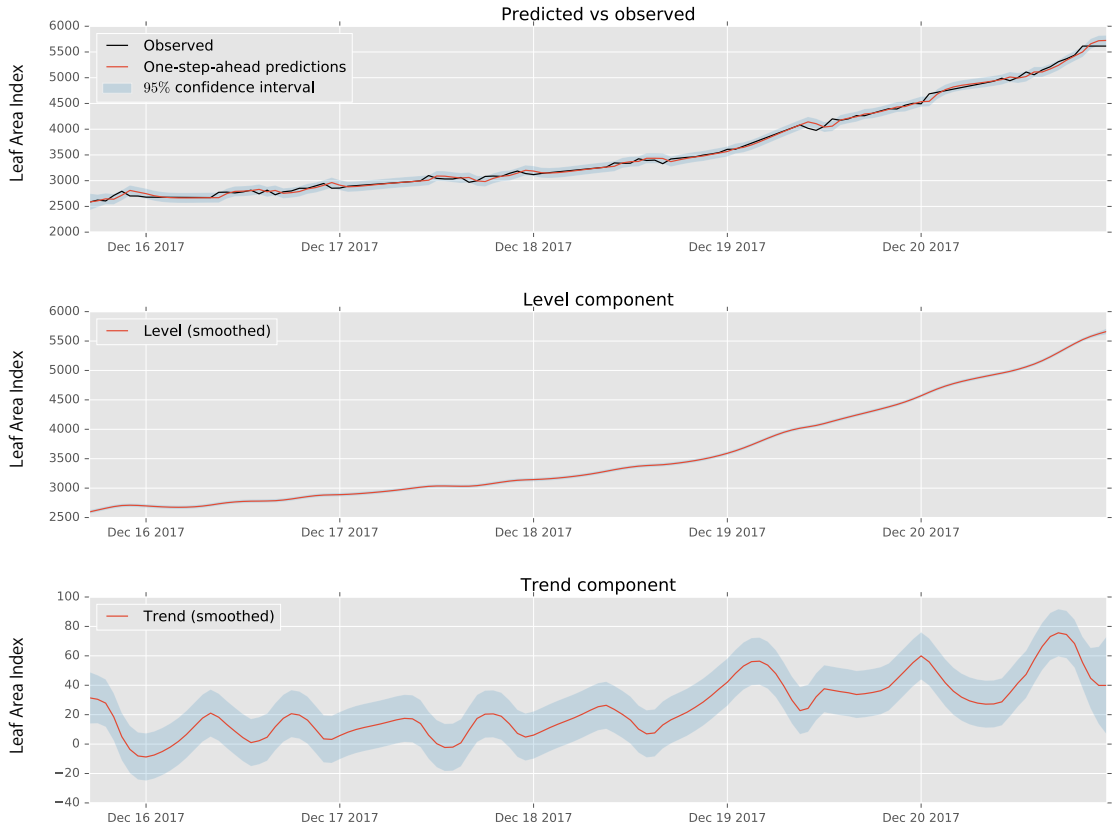

図 4.5: 葉面積の観測値と平滑化した値の時間遷移グラフ.

4.2.4　一連の手続きのまとめ

4.2.2 節および 4.2.3 節で述べたデータ処理を行って，実験期間中に変化していく同一の株の葉面積の値を含んだ系列データを得たい．しかし，1. 葉面積取得プログラムは，入力画像によっては失敗し間違った値を出力することがある，2. 異なる実験期間において収集したデータは異なる系列データとしてまとめたいが，ファイル名からは系列を区別できない，という 2 つの問題があるため，画像解析のハイパーパラメータ管理や実験時期による画像のラベル付けなどを順序立てて行う必要がある．本節では，サーバ上に置かれた画像ファイルの一覧から葉面積の系列データとしてまとまるための処理手順を示す．

テーブル設計

まず，異なる実験期間の葉面積のデータを異なる系列として区別したい．そのために各実験期間に “growth_id” として番号を割り当てたい．growth_id は全ての実験期間を一意に特定できるべきであり，異なる Personal Food Computer で行われた栽培であっても重複は許さないようにする．この growth_id を定義するマスターテーブルとして表 4.1 に示す growth_id_master.csv を作成した．実験を行うたびにこの csv ファイルに実験期間に関

する情報を 1 行加えれば良い．

表 4.1: growth_id_master.csv の各フィールドの説明．

カラム名	値の意味
pfc_id	実験を行った PFC の id．
growth_id	PFC における実験の id．
from	実験期間の開始日時．
to	実験期間の終了日時．

また，画像処理プログラムにはいくつかのパラメータを要することが多く，その値によって葉面積値の計算の成功可否が異なる．そこで，パラメータのまとまりを区別するためのマスターテーブルとして，表 4.2hyper_param_master.csv を作成した．最初のカラムが id として，残りのカラムパラメータの各値としての役割を持っている．

表 4.2: hyper_param_master.csv の内容．最初のカラムが hyper_param を指定する id として表され，残りのカラムが hyper_param の値に関する情報である．

hyper_param_id	bin_threshold	fill_threshold	kernel_size	roi-height	roi-width	roi-x	roi-y	rotate_angle
1	120	2500	11	-170	-450	400	100	45
2	140	2500	11	-170	-450	400	100	45

この 2 つのマスターファイルを事前に作成すれば，次の節で述べる処理を行うことで葉面積の系列データがまとまった形で求まる．

データ処理のためのスクリプト

本節では，4.2.4 節で作成したマスターファイルと 4.2.2 節および 4.2.3 節で述べたプログラムを用いて平滑化された時系列の葉面積データを取得する一連の手順を述べる．

img_proc_master.csv の作成

growth_id_master.csv と hyper_param_master.csv を入力とし，解析するべき画像のファイルパスおよび画像解析に使用するパラメータ id を割り当てたテーブル img_proc_master.csv を生成する．テーブルのカラムに関して表 4.3 に示す．

表 4.3: img_proc_master.csv の各フィールドの説明．

カラム名	値の意味
timestamp	画像取得を行った日時．
hyper_param_id	画像解析に使用するパラメータの id．
file_path	画像処理を行う対象のファイルへのパス
pfc_id	実験を行った PFC の id．
growth_id	PFC における実験の id．

raw_table.csv の作成

次に，生成された img_proc_master.csv を用いて各株に関する情報を格納するテーブルを作成する．テーブル名を raw_table.csv として作成し，各カラムについて表 4.4 に述べる．raw_table.csv の生成は以下のコマンドで実行される．

```
$ python make_raw_table.py
```

表 4.4: raw_table.csv の各フィールドの説明．

カラム名	値の意味
timestamp	画像取得を行った日時
file_path	画像処理を行う対象のファイルへのパス
pfc_id	実験を行った PFC の id
growth_id	PFC における実験の id
pos	実験期間中における株の位置
plant_id	株を一意に指定する id
contour_area	株の葉面積値
moment_x	重心の x 座標
moment_y	重心の y 座標
error_type	画像処理時に生じたエラーの種類
hyper_param_id	画像解析に使用するパラメータの id

lai.csv の作成

生成された raw_table.csv より系列毎の葉面積を格納したテーブルを取得したい．そこで，以下のコマンドを実行することによりテーブル lai.csv を生成する．

```
$ python make_lai.py
```

lai.csv の各カラムについては表 4.5 に示す．“bottom”，“left”，“right”，“top” は取得画像の例 4.3 における各位置に配置された株の葉面積を示す．

表 4.5: lai.csv の各フィールドの説明．

カラム名	値の意味
timestamp	画像取得を行った日時
bottom	画像 4.3 下に設置された株の葉面積
left	画像 4.3 左に設置された株の葉面積
right	画像 4.3 右に設置された株の葉面積
top	画像 4.3 上に設置された株の葉面積
growth_id	画像解析に使用するパラメータの id

filtered_lai.csv の作成

生成された lai.csv に表れる葉面積のデータには欠損値が多かったり，ノイズが大きかったりなどの理由からデータ解析に直接扱いにくい場合が多い．そこで lai.csv を受取り，平滑化し，等間隔の葉面積時系列を出力するスクリプト filter_lai.py を実装した．filtered_lai.csv には，各株にたいして葉面積の平均と分散の値の時系列値が格納されている．

4.3　スコア抽出

4.3.1　スコアの定義

レシピの良し悪しを定量的に評価するために，取得される A, E, S から一意に決定されるスカラ値をスコアとして用意する．これにより最適なレシピとはスコアを最大化するレシピであり，レシピ探索問題をバンディットアーム問題として定義できる．スコアは一回の栽培における最終時刻で得られるあたいでも良いし，時々刻々の A, E, S から得られる時系列の値でも良い．それぞれのスコアの取得方法のコンセプトを図 4.6 に示す．レシピの探索を早めるという観点では，一回の栽培で一つのスコアしか求められないことは効率が悪いため，本稿では主に時系列のスコアに着目する．次節に活用可能なスコアの例を取り上げる．

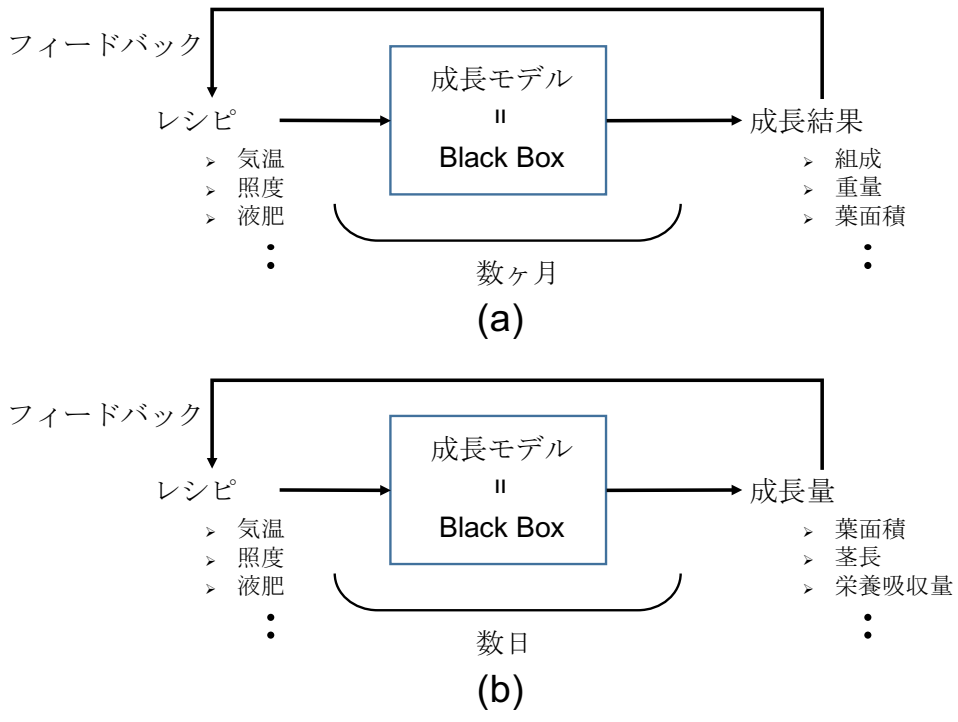

図 4.6: スコアの概念図．(a) は一回の生育につき一つのスコアを得られるモデルを表し，(b) は一回の生育中に連続的に成長量を観測することで得られるスコアが複数回訪れるモデルを表す．

4.3.2　スコアの例

前節ではレシピの良し悪しを定量的に評価するためにスコアを決定することが重要であることを述べた．以下に，具体的に定義可能なスコアとしてどのようなものがあるかを記す．

成長率

レシピの良し悪しを評価する指標の一つに収量がある．日々の植物体の様子は天井のカメラにより観測可能であるため，葉面積を最大化させるようにレシピの最適化を行いたい．しかし，面積は指数関数的に増加するためスコアとして採用はできない．そこで，指数的成長を考慮した成長率をスコアとして採用することは可能である．成長率をスコアとした場合の探索手法に関しては 5 で詳述する．

光合成速度

植物は光合成を通じて体を大きくするため，光合成の活動量はスコアとして設定可能である．4.2.2 節でも述べたように，CO2 濃度の変化から光合成速度は求められるため，これを最大化するスコアを求めれば良い．

成長率 + 稼働コスト

上記二つは植物状態のみを考慮したスコアであったが，実際の植物工場においては様々な制約条件下での最適なレシピを求めたい．制約条件として考えられうるものの一つは稼働コストであろう．成長率に正の係数，稼働コストに負の係数を掛け合わせた和をスコアとして定義すれば，稼働コストを制約としたレシピの最適化が図れる．

4.4　動的レシピ変更

4.4.1　概要

図 4.6(b) に示すレシピ改良において，日々の成長量とレシピの関係性から翌日のレシピを賢く選択する必要がある．レシピ選択において機械学習手法を適用した場合計算量が大きくなるため，多くの ROS ノードを Raspberry Pi 上で動作させている PFC 上で計算を行うことは適さない．そこで，3.4.8 節でも述べた，PFC と同一 LAN 内にある仮想マシン (VM) において翌日のレシピを計算し，VM から動作中の PFC において稼働レシピの内容を追加する機構を実装した．

4.4.2　レシピ更新のためのシステム構成

VM 上で計算したレシピを ROS を稼働している PFC において動的にレシピを追加するシステムを構築したので，システムの概要図を図 4.7 に示す．

レシピを更新するための機構は，VM 上に立てた HTTP サーバおよび PFC 上で稼働する ROS ノード recipe_update より構成される．

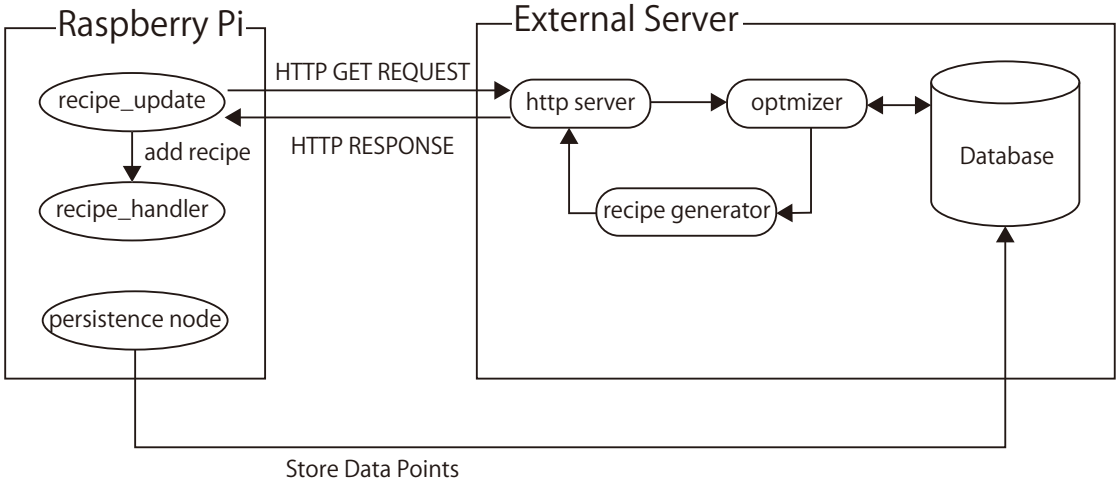

図 4.7: VM から PFC が実行しているレシピを動的に変更するためのシステム構成図.

HTTP サーバ

VM は PFC より HTTP GET リクエストを受け，翌日のレシピの情報を含めたオブジェクトを JSON 形式で返す．JSON のプロパティについて表 4.6 に示す.

表 4.6: VM から返される JSON の各プロパティ

カラム名	値の意味
phase	レシピの内容を表現する phase.
update_interval	PFC が次にレシピに更新を依頼するまでの時間.
phase_duration	phase を全て実行するのに要する時間.

HTTP サーバは図 4.7 に示すとおり，server, optimizer, recipe_vector の 3 つのプログラムより構成される．server は 8000 番ポートを監視し，GET リクエストが来た時にパラメータを optimizer に渡す．GET リクエストによってパラメータ search_func が必ず渡される．search_func は optimizer のどの関数を用いて最適化を行うかを指定し，GET リクエストの他のパラメータはこの実行される関数の引数として渡される．optimizer は PFC のセンサ情報を格納したデータベースから必要なデータを取り出し，スコアを考慮して次に選択するべきレシピを計算する．optimizer が出力するレシピのフォーマットは 3.4.5 に示したフォーマットとは異なり，要素を 12 個持つ 1 次元ベクトルとしたものである．recipe_vector は optimizer が出力した 1 次元ベクトルをレシピのフォーマットに変換するためのプログラムである.

4.4.3　recipe_upadte によるレシピの内容書き換え

PFC においてレシピの管理は ROS ノード “recipe_handler” が行っており，実行するレシピの内容などはこのプロセス以外からは変更できない．そこで，レシピ更新に関して外

部とのインターフェースとしての役割を持つ recipe_update を実装し，recipe_update からレシピデータを受け取った時に recipe_handler が実行するレシピの内容を更新するように改良した．recipe_update は VM から受け取る表 4.6 の JSON データの update_interval および phase_duration の値をもとに次に HTTP リクエストを送るタイミングを管理する役割を持つ．

第5章

成長率モデルを利用したベイズ最適なレシピ探索手法

5.1　概要

最適レシピの発見に向けた研究は多く行われてきたが，そのほとんどは得られた環境センサ値と環境応答の時系列データを持っている前提で予測精度の高いモデル構築をし．所望の環境応答を得るようなレシピを逆算して求めるという手法であった．これは予測モデルがレシピの探索範囲で常に高い精度をもつことで初めて有効となる手法であるが，リカレントニューラルネットワークなどの自由度の高い時系列モデルなどを用いる場合は，学習するべきレシピの組み合わせの数が膨大となってしまい，データの収集コストの高い植物には不適である．また，多くの研究は学習データの収集方法については言及していなかった．そこで本研究では，必要となる学習データ数を少なくするために，葉面積の即日の成長率がコンテキスト依存せずレシピによって一定であるという単純なモデルを仮定することで，レシピを稼働したときの報酬が即日で得られるバンディット問題とみなせるようにし，ベイズ最適化を用いた探索と活用のトレードオフを考慮したレシピ探索手法を提案する．

図 5.1 に示す状態空間モデルを仮定し，レシピ a と成長率 r がガウス過程に従う非線形な関数 F によって植物の成長量を表現できると仮定した．これにより，時々刻々得られるノイズ付きの関数 F にベイズ最適化を用いて，信頼上限を最大化するレシピが求められ，最適レシピの発見が早まる．

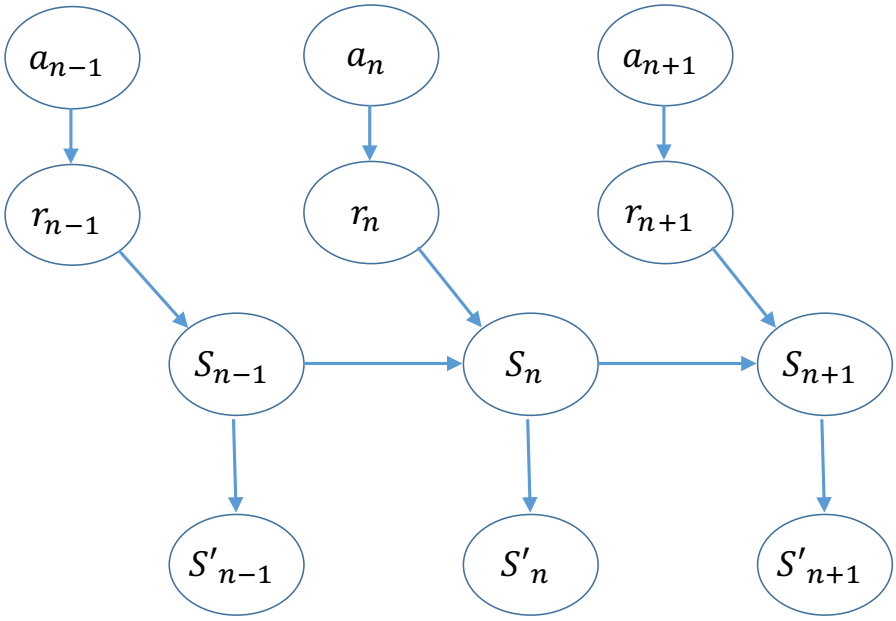

図 5.1: 成長率がレシピによって一定であるという仮定をおいたときの，観測される葉面積値の状態空間モデル．

5.2　成長率モデル

5.2.1　成長率

植物は主に光合成を通じて成長し，葉が大きくなるほど光合成量も増加するため，栄養成長期には指数的成長を示す．この現象は Blackman のモデルとして以下の式で表される．

$$\frac{\mathrm{d}S}{\mathrm{d}t} = r \cdot S \tag{5.1}$$

ここで S は葉面積，r は成長率を表し，一般的に成長率は環境や生育ステージによって変化する．

5.2.2　成長率に着目した状態空間モデル

レシピの探索を高速に行うために，成長率がレシピの関数とし式 (5.2) および式 (5.3) によって表されるモデルを提案する．S_t は時刻 t における葉面積の真の値，S'_t は時刻 t における葉面積の観測値を表す．σ^2 は観測によって生じる誤差を分散として表し，正規分布と仮定する．ここで時刻 t の成長率をレシピのベクトル表現である a_t を変数とする関数 $F(a_t)$ によって表現できると仮定した．これによって日毎のレシピに対して観測される成長率を観測しながら，次に試すレシピを賢く選択することで，最適なレシピをより早く発見できる．

$$S'_{t+1} \sim \mathcal{N}(S_{t+1}, \sigma^2) \tag{5.2}$$

$$\frac{S_{t+1} - S_t}{\Delta t} = F(\boldsymbol{a}_t) \cdot S_t \tag{5.3}$$

5.3　ベイズ最適化を用いた動的レシピ選択手法

式 (5.2) および式 (5.3) で定義したモデルは，観測される成長率を報酬として，報酬を最大とするように翌日のレシピを選択する連続腕バンディット問題とみなせる．ここで関数 F がガウス過程に従うことを仮定すれば，信頼上限を最大とするレシピを常に選択する，ベイズ最適化を用いて効率的な探索が可能となる．

報酬として観測された成長率の集合 $\boldsymbol{r}_t = \{r_1, r_2, ..., r_t\}$，レシピベクトルの集合 $\boldsymbol{a}_t = \{a_1, a_2, ..., a_t\}$ があるとき，$F(a_t)$ の信頼上限 $\overline{\mu_a(t)}$ は以下の式で求められる [34]．

$$\overline{\mu_a(t)} = \mu(a|\boldsymbol{r}_t) + \alpha_t \cdot \sigma(a|\boldsymbol{r}_t) \tag{5.4}$$

ここで，$\mu(a|\boldsymbol{r}_t)$ は成長率 $\boldsymbol{r}_t$ が観測されたときにレシピ a を選択したときに得られる成長率の期待値を表し，$\sigma(a|\boldsymbol{r}_t)$ は分散を表す．α_t は信頼度を表すパラメータである．$\mu(a|\boldsymbol{r}_t), \sigma(a|\boldsymbol{r}_t)$ は関数 F がガウス過程に従うと仮定したとき，$\boldsymbol{r}_t, \boldsymbol{a}_t$ より求められる．各時刻 t において $\overline{\mu_a(t)}$ を最大化するレシピ a_t を選択することで，ベイズ最適な探索が可能となる．これは Gaussian Process UCB 方策（GP-UCB 方策）と呼ばれる．

5.4　ガウス過程における予測分布の更新

5.3 節では，$\mu(a|\boldsymbol{r}_t), \sigma(a|\boldsymbol{r}_t)$ が求まった上で，次に選択するべきレシピを求める手法を述べた．本節では観測された成長率の集合 $\boldsymbol{r}_t$，適用したレシピの集合 $\boldsymbol{a}_t$ があるときに $\mu(a|\boldsymbol{r}_t), \sigma(a|\boldsymbol{r}_t)$ を求める計算方法を述べる．

まず，関数 F が不連続である，もしくは任意の傾きで急激に変化することを許した場合には有限個のデータでは，F 及び最適なレシピ a_t^* の推定は難しくなる．そこでガウス過程においては，2 つのレシピ (a, a') が近ければ，$F(a), F(a')$ も近い値を示すことを定量的に示す共分散関数 $k(a, a')$ を用いている．この共分散関数によって関数の滑らかさに制約を与える．共分散関数は以下のように表される．

$$k(a, a') = \sigma_0^2 g(\|a - a'\|_{\boldsymbol{\lambda}}) \tag{5.5}$$

ここで，$\|a\|_{\boldsymbol{\lambda}} = \sqrt{\sum_{i=1}^{d}(a_i/\lambda_i)^2}$ はレシピ a のスケールパラメータ $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_d) \in (0, \infty)^d$ で規格化したユークリッドノルムであり，σ_0 は F のスケールパラメータである．また，関数 g はカーネル関数を表し，最も代表的なガウスカーネルは以下の式で表される．

$$g(z) = \exp(\frac{-z^2}{2}) \tag{5.6}$$

そこで，$F(\boldsymbol{a}_t) = (F(a_1,), F(a_2,), ..., F(a_t)) \in \mathbb{R}$ とするとき，定義された共分散関数 $k(a, a')$ を用いて，関数 F がガウス過程に従うことは $F(\boldsymbol{a}_t)$ が多変量正規分布に従うことをいい，以下のように表現できる．

$$F(\boldsymbol{a}_t) \sim \mathcal{N}(\mu(\boldsymbol{a}_t), k(\boldsymbol{a}_t, \boldsymbol{a}_t)) \tag{5.7}$$

ただし，

$$\mu(\boldsymbol{a}_t) = (\mu(a_1), \mu(a_2), ..., \mu(a_t)) \tag{5.8}$$

$$k(\boldsymbol{a}_t, \boldsymbol{a}'_s) = \begin{pmatrix} k(a_1, a'_1) & k(a_1, a'_2) & \dots & k(a_1, a'_s) \\ k(a_2, a'_1) & k(a_2, a'_2) & \dots & k(a_2, a'_s) \\ \vdots & \vdots & \ddots & \vdots \\ k(a_t, a'_1) & k(a_t, a'_2) & \dots & k(a_t, a'_s) \end{pmatrix} \tag{5.9}$$

まだデータセットがない状況など，関数 F に関する事前知識がない場合は $\mu(a) = 0$ とすればよい．ここで，実際に観測された成長量 $\boldsymbol{r}_t$ が多変量正規分布 $\mathcal{N}(\mu(\boldsymbol{a}_t), k(\boldsymbol{a}_t, \boldsymbol{a}_t) + \sigma^2 I_t)$ に従い，取りうるレシピ $\boldsymbol{a}'_s = \{a'_1, a'_2, ..., a'_t\}$ の報酬期待値 $F(\boldsymbol{a}'_s)$ との事前同時確率密度は

$$\boldsymbol{z}_{t,s} = (\boldsymbol{r}_t - \mu(\boldsymbol{a}_t), F(\boldsymbol{a}'_s) - \mu(\boldsymbol{a}'_s)) \tag{5.10}$$

とおいたとき，以下のように与えられる．

$$\pi(\boldsymbol{r}_t, F(\boldsymbol{a}'_s)) \propto \exp\left(-\frac{1}{2}\boldsymbol{z}_{t,s}\begin{pmatrix} k(\boldsymbol{a}_t, \boldsymbol{a}_t) + \sigma^2 I_t & k(\boldsymbol{a}_t, \boldsymbol{a}'_s) \\ k(\boldsymbol{a}'_s, \boldsymbol{a}_t) & k(\boldsymbol{a}'_s, \boldsymbol{a}'_s) \end{pmatrix}^{-1} \boldsymbol{z}_{t,s}^{\mathrm{T}}\right) \tag{5.11}$$

そこで $K_t = k(\boldsymbol{a}_t, a_t) + \sigma^2 I_t$ とし，式 (5.11) を変形すれば，成長率の集合 $\boldsymbol{r}_t$ を受け取ったときの次に実行可能なレシピの集合 $\boldsymbol{a}'_s$ の期待値 $F(\boldsymbol{a}'_s)$ の事後同時分布は平均と分散がそれぞれ以下の式となることがわかる．

$$\mu(\boldsymbol{a}'_s|\boldsymbol{r}_t) = \mu(\boldsymbol{a}'_s) + (\boldsymbol{r}_t - \mu(\boldsymbol{a}_t))K_t^{-1}k(\boldsymbol{a}_t, \boldsymbol{a}'_s) \tag{5.12}$$

$$\sum(\boldsymbol{a}'_s|\boldsymbol{r}_t) = k(\boldsymbol{a}'_s, \boldsymbol{a}'_s) - k(\boldsymbol{a}'_s, \boldsymbol{a}'_t)K_t^{-1}k(\boldsymbol{a}_t, \boldsymbol{a}'_s) \tag{5.13}$$

また，K_t に関して

$$K_{t+1} = \begin{pmatrix} K_t & k(\boldsymbol{a}_t, a_{t+1}) \\ k(\boldsymbol{a}_{t+1}, a_t) & k(a_{t+1}, a_{t+1}) + \sigma_0^2 \end{pmatrix}^{-1} \tag{5.14}$$

として逐次的に計算が可能となる．つまり，次に選択するレシピ a にたいして，予測分布の平均と分散が次のように求まる．

$$\mu(a|\boldsymbol{r}_t) = \mu(a) + (\boldsymbol{r}_t - \mu(\boldsymbol{a}_t))K_t^{-1}k(\boldsymbol{a}_t, a) \tag{5.15}$$

$$\sigma^2(a|\boldsymbol{r}_t) = k(a, a) - k(a, \boldsymbol{a}'_t)K_t^{-1}k(\boldsymbol{a}_t, a) \tag{5.16}$$

この平均と分散が求まったとき，5.3 節で述べたように，GP-UCB 方策におけるベイズ最適なレシピ a^* は

$$a^* = \arg\max_a \mu(a|\boldsymbol{r}_t) + \alpha\sigma(a|\boldsymbol{r}_t) \tag{5.17}$$

として求まる．ここで α は信頼度を表すパラメータであり，大きいほど未知のレシピを探索し，小さいほど既に得られた知識を活用するようになる．

5.5　レシピと成長率の関係性の測定

成長率とレシピの関係性を明らかにするために，簡易型の植物工場である Personal Food Computer [27] (PFC) を用いてレタスの生育実験を行った．実験期間を，植物の葉長が 3 cm 以上 15 cm 以下である栄養生長期に限定することで，生育ステージの違いによる成長率への影響を少なくした．また、PFC への移植時点での各個体の葉面積差が成長率にどのように寄与するかも検討できるように実験対象とする個体の選定を行った．PFC 内で撮影された植物体の画像の例を図 4.3 に示す．カメラは 1 時間おきに撮影し，毎時の葉面積データを取得できる．平滑化した 1 時間ごとの葉面積の時系列グラフを図 5.2 に示す．1 回転につき 4 株の生育を行うため，4 系列 (“top”, “left”, “right”, “bottom”) の葉面積の時系列データを示している．実験は 12 月 15 日から 12 月 20 日の 6 日間にかけて行った．適用したレシピの詳細を表 5.1 に示す．

次に，図 5.2 のデータと式 (5.1) を用いて成長率の算出を行った．式 (5.1) における $\mathrm{d}t$ はレシピのサイクルである 24 時間として，各時刻において 24 時間後の葉面積との比較から成長率を算出した．4 株の成長率の平均をヒストグラムとして図 5.3 に示す．実験期間における成長率の平均は 3.1×10^{-6} s^{-1}，分散は 4.7×10^{-7} s^{-1} となった．平均に対して分散の値は小さく，正規分布として仮定可能だと言える．

表 5.1: 実験期間中に適用したレシピの環境制御の各値．昼夜を交互に繰り返す．

	時間	照明	気温	pH	換気間隔
昼	17 時間	ON	23 °C	6.0	2 時間
夜	7 時間	OFF	21 °C	6.0	2 時間

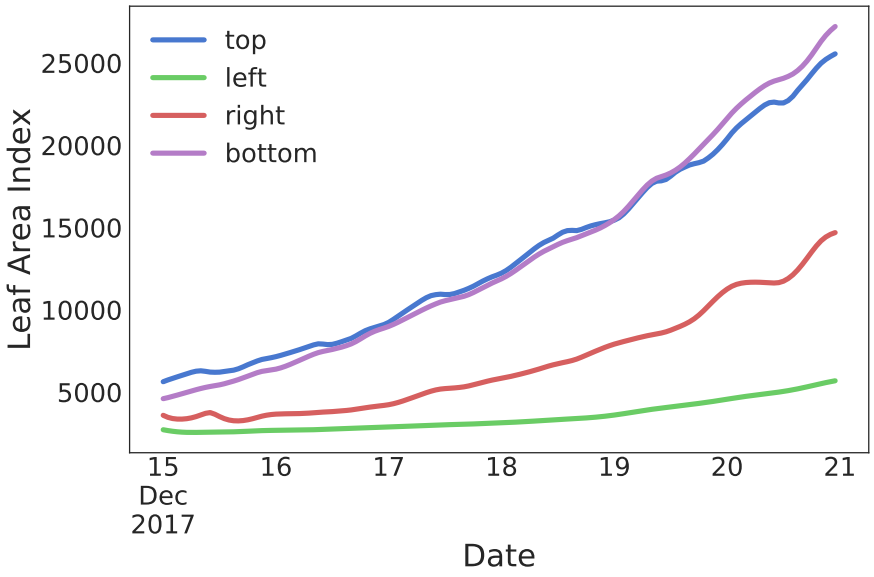

図 5.2: 実験期間における葉面積の時系列グラフ．’top’, ’left’, ’right’, ’bottom’ は図 4.3 におけるそれぞれの位置の植物体を表す．

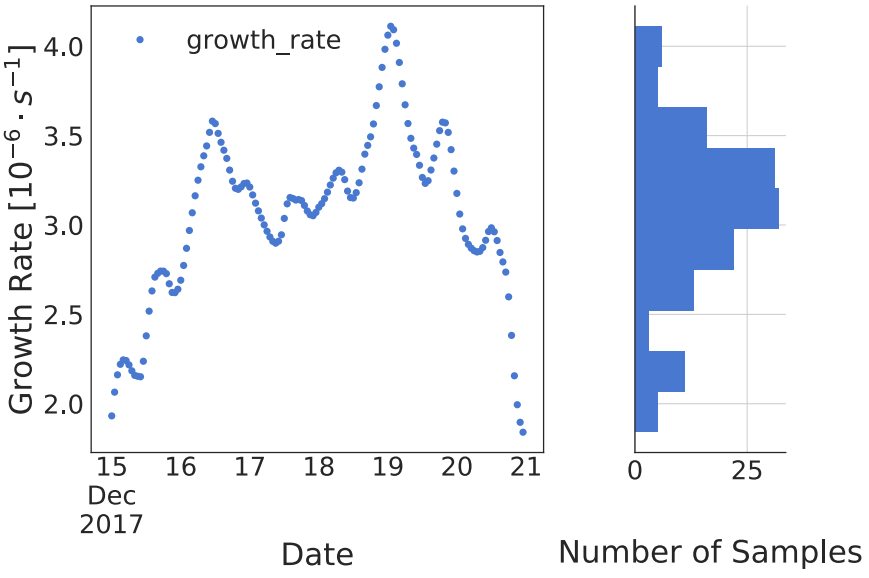

図 5.3: 実験期間においてレシピ一定の下，1 時間ごとに算出した成長率の時間遷移グラフおよびヒストグラムを示した．

次にレシピを 1 日ごとに変更したときの成長率を観測した．表 5.1 における昼の時間のみを変更させていった．2018 年 1 月 18 日の 17 時より 24 時間ごとに昼の時間を，12 → 16 → 20 → 24 → 12 → 16 の順番で変更していった．この条件下で観測された成長率の遷移を図 5.4 に示す．垂直方向の罫線はレシピが変更された時刻を示す．レシピの変更間隔と成長率の算出に用いる時間間隔は 24 時間で等しいため，レシピによって成長率に変化が生じれば，グラフは各日の成長率の変化の移動平均として表されることになる．図より，19 日 → 20 日 → 21 日 → 22 日 と成長率が増加していることが分かる．一方で 23 日に関しては 22 日と比較して成長率が減少しているものの，同じレシピを実行した 19 日と比較すると値が大きくなっていることが分かる．これは成長率にも前時刻の勢いが影響している可能性が高い．時刻間の影響が生じないパラメータとして成長率を選んだが，前時刻に影響されないパラメータであるとはこの結果からは言い難い．

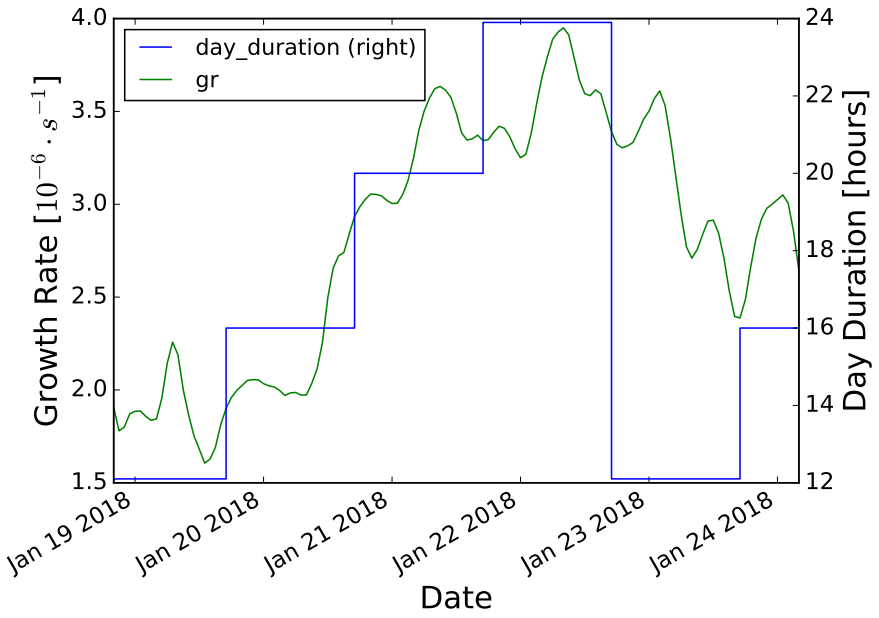

図 5.4: 実験期間において 1 日ごとにレシピを変更したときの 1 時間ごとに算出した成長率の遷移図を示した．

5.6　各レシピの成長率の算出

毎時刻の成長率の算出にあたり，24 時間後の葉面積と比較し式 (5.1) を用いたことを前節でのべた．比較する 2 点の時間間隔を 24 時間としたのは，レシピの適用時間が 24 時間であるためである．しかし，レシピの評価をするために求めたいものは各レシピを示す 1 次元の成長率であり，24 時間ごとにレシピを変更していった時に観測される毎時間の成長率から算出する必要がある．そこで，レシピ A とレシピ B をそれぞれ 24 時間適用し，24 時間ごとの比較計算によって求まる成長率の集合 $R_{day} = \{r_0, r_1, ..., r_24\}$ が得られている状況を考える．求めたいものは，レシピ A の成長率 r_a 及びレシピ B の成長率 r_b である．

しかし，これらの成長率はレシピ A 及びレシピ B の両方の影響を受けたものであり，また成長率の時刻によっても各レシピの寄与の度合いが異なる．

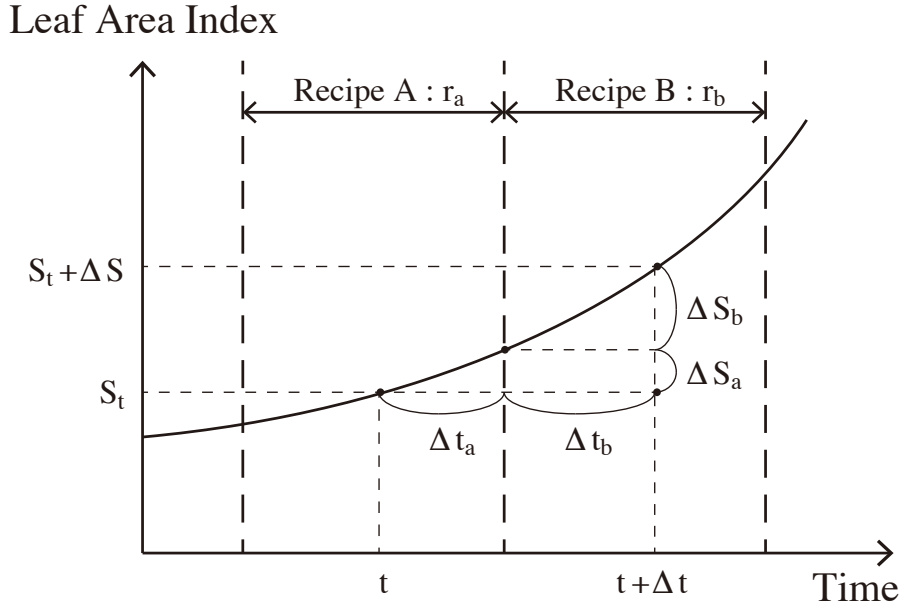

図 5.5: 成長率の算出に必要とした期間に 2 つの異なるレシピが適用されているときの，各レシピの成長率の計算方法に関する図．

そこでまず，異なる 2 つのレシピの影響を受けた成長率 r から r_a，r_b を求めたい．図 5.5 に示すように，レシピ A，レシピ B の適用時間を t_a，t_b とし，レシピ A，レシピ B の各適用時間における葉面積の増加分を ΔS_a，ΔS_b，そして $\Delta S = \Delta S_a + \Delta S_b$，$\Delta t = \Delta t_a + \Delta t_b$ とすると，式 (5.1) を用いて以下の関係が求められる．

$$\Delta S = r \cdot \Delta t \cdot S_t \tag{5.18}$$

$$\Delta S_a = r_a \cdot \Delta t_a \cdot S_t \tag{5.19}$$

$$\Delta S_b = r_b \cdot \Delta t_b \cdot (S_t + \Delta S_a) \simeq r_b \cdot \Delta t_b \cdot S_t \tag{5.20}$$

式 (5.18) の左辺に式 (5.19)，式 (5.20) を代入すると以下の式が成り立つ．

$$r_a \cdot \Delta t_a + r_b \cdot \Delta t_b = r \cdot \Delta t \tag{5.21}$$

これを R_{day} に適用すれば，各時刻に対して式 (5.21) が成り立つために 24 個の連立方程式ができる．成長率には多くの誤差が乗っているために，全ての方程式を満たす解 r_a，r_b が存在する可能性は低い．そこで，R_{day} の元 r_n に対するレシピ A，レシピ B の適用時間を t_{na}, t_{nb} とすると，最小 2 乗法によって r_a，r_b は次のように求められる．

$$r_a = \frac{T_{ab} \cdot K_b - T_{bb} \cdot K_a}{T_{ab} - T_{aa} \cdot T_{bb}} \tag{5.22}$$

$$r_b = \frac{T_{ab} \cdot K_a - T_{aa} \cdot K_b}{T_{ab} - T_{aa} \cdot T_{bb}} \tag{5.23}$$

ただし，$T_{aa}, T_{bb}, T_{ab}, K_a, K_b$ は以下の値である．

$$T_{aa} = \sum \Delta t_{na}^2, \quad T_{bb} = \sum \Delta t_{nb}^2, \quad T_{ab} = \sum \Delta t_{na} \cdot \Delta t_{nb},$$
$$K_a = \sum r_n \cdot \Delta t_n \cdot \Delta t_{na}, \quad K_b = \sum r_n \cdot \Delta t_n \cdot \Delta t_{nb}$$

この計算方法を図 5.6 に示したデータに適用することで，計算したレシピ毎の成長量，図 5.6 に示す．日照時間が等しければ成長量も等しくなるモデルを仮定したが，1 月 18 日と 22 日の成長率を比較すれば分かるように，日照時間が共に 12 時間の 2 つのサンプル点で大きな差が生じていることが分かる．22 日の場合では前日に適用したレシピの照射時間が 24 時間で高い成長率を示しており，その時間遅れの応答としてレシピの成長量を増加させていると考えられる．レシピと成長率を 1 対 1 に対応するモデルを前提としたが，この観測結果からはモデルの妥当性を示すのは難しい．しかし，ガウス過程において観測される誤差を大きく見積もればベイズ最適化の適用は精度は落ちるが可能である．次節ではこの観測結果にベイズ最適化を適用した結果を述べる．

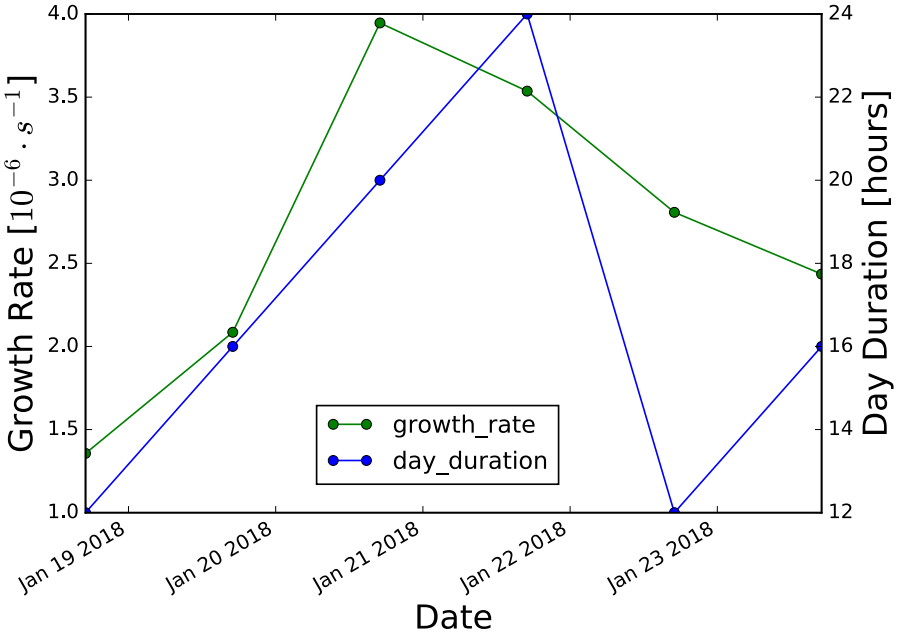

図 5.6: 1 時間ごとの成長率から算出した日毎の成長率と対応するレシピの日照時間の遷移グラフ．

5.7　ベイズ最適化の適用結果

5.6 節で述べた手法により得られた各レシピに対応する成長率のデータを用いて，関数がガウス過程に従うとしたときの予測分布及び信頼区間を計算した．その結果を図 5.7 の上段に示す．横軸が日照時間，縦軸が成長率を表しており，事後分布（Prediction）は日照時間が高いほど成長率も高くなる傾向を示唆している．全体のサンプル数は 6 と少ない

にも関わらず，照明時間が長いほど成長率が高いという植物の代表的な活動である光合成による成長の特徴を捉えることが出来た．

また，式 (5.4) における α を 0.01 として，活性化関数の値 $\overline{\mu_a(t)}$ をプロットしたものが下段に示されている．これを最大にするものが次に設定するべき日照時間の値であり，このグラフによれば日照時間が 8.0 時間レシピを次に選択すればよいことがわかる．これはバンディット問題における「探索」の要素を取り入れ，サンプル数の少ない未知のレシピを試すという意図から算出された値である．このようにベイズ最適化を用いれば，既に得られたデータの活用によって期待値が高いレシピを選択するのみならず，サンプル数が少なく未知なレシピを積極的に試すことで効率的な探索が可能となる．今後は，日照時間のみならず複数の環境要因が変化する状況下でのベイズ最適化の適用を行い，手法の正当性の検討が望まれる．

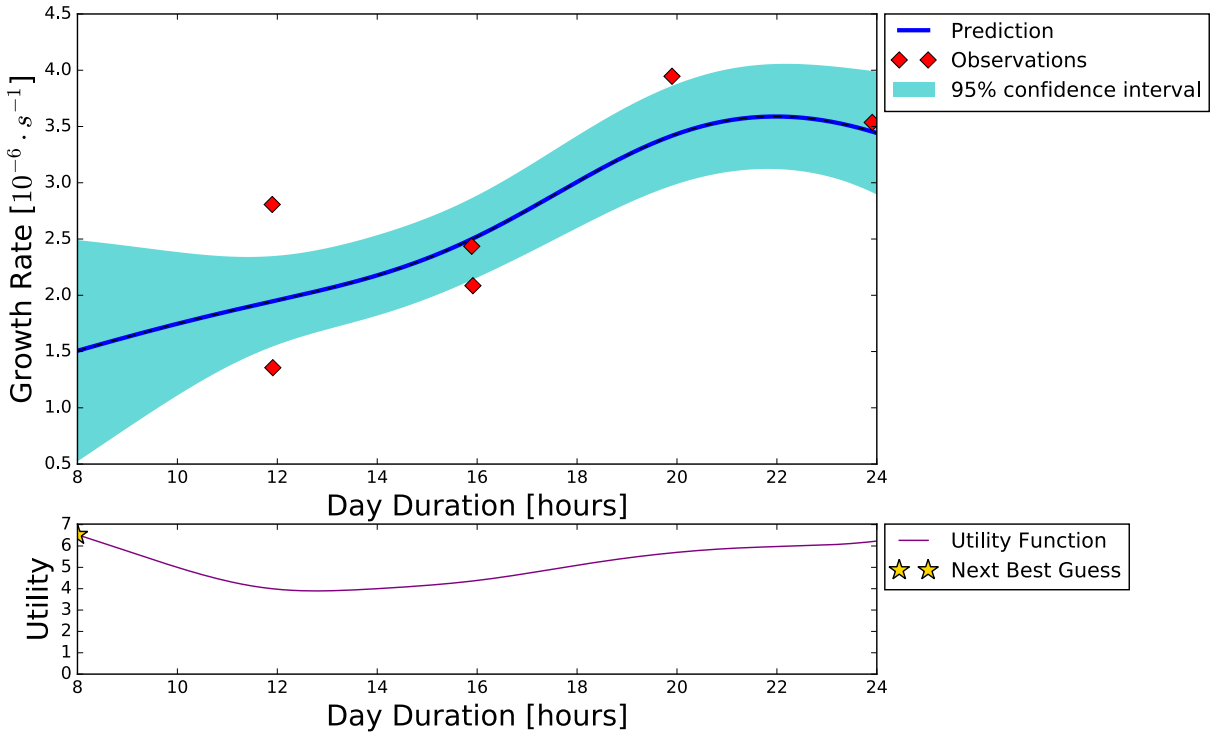

図 5.7: ガウス過程に従うことを仮定したときの成長率の観測点と予測分布，信頼区間を示したグラフ（上段）．下段は式 5.4 に示した活性化関数の値を示しており，これを最大とする日照時間を次のレシピとして選択することでベイズ最適な探索が可能となる．

第6章

結論

本稿では，植物工場における最適なレシピをデータドリブンで発見するための手法として，即日の葉面積の成長率がレシピによって決まるという単純なモデルを仮定することで成長率を報酬としたバンディット問題へと帰結し，ベイズ最適化を用いる手法を提案した．結果として，即日の成長率は以前の成長率や環境の影響を受け，提案した成長率のモデルでは高い精度で表現できないことが確認された．しかし，ベイズ最適化はある程度の誤差も許容できるため，ある程度の誤差を伴うガウス過程に従うレシピの関数によって成長率を表現できると仮定した時，日照時間が多いほど成長率も高くなるという予測分布を示した．これは植物の最も重要な活動の一つである光合成の特性を捉えている．ニューラルネットワークなどの表現力の高い機械学習モデルでは大量のデータを必要とするが，提案する手法は数十個程度の少ないサンプル数でも有用な知見を示すことが可能であり，データの収集コストが高い植物に適した手法であることが分かった．

また，完全制御型の植物工場である Personal Food Computer (PFC) 及びそこで収集されたデータを解析する基盤を一から実装し作り上げた我々の経験に基づき，レシピ探索を行う上での実装されるべきハードウェア及びソフトウェアの要件を 3 章および 4 章で抽象的に述べた．これは，PFC をこれから作成する人のみならず様々な植物工場においてレシピの最適化を行って植物工場の生産性を高めたいと思う人に対して有意義なものとなると考えられる．今後の方針としては，葉面積の状態空間モデルにおいて，前時刻のレシピや状態からの影響を受けず即日のレシピによって決定される変数で構成される状態空間モデルを提案し，高い精度で表現できることを検証する．精度の高い状態空間モデルが提案されれば，5 章で提案したベイズ最適化を用いて，効率のよいレシピ探索が可能となるであろう．

謝辞

本研究ならびに修士論文の執筆にあたり，研究の本質やポイント，見せ方など研究にあたって重要なことをご教授下さった川原圭博准教授に，深く感謝いたします．また，修士 1 年次に，鋭いコメント及び指導で研究内容に関して多くの気づきをもたらし，その他にも人生観や仕事に対する考え方についても教えてくださった浅見先生に，深い感謝を申し上げます．そして，Personal Food Computer (PFC) の組み立てのための工作法や研究方針，論文校正など多大なるサポートをしてくださった繁田さんにも大変お世話になり，誠にありがとうございます．ならびに，研究室の先輩である高木さん，鳴海さん，奥谷さん，Tung さんには数多くの指導をいただき，感謝いたします．

その他にも，和やかな話で研究室の雰囲気を常に明るくしてくれた池内くん，データ解析の作業を多分に手伝ってくれた中原くん，PFC の組み立てを手伝ってくれた鈴木くん，石毛くん，角谷くん，高橋くん，毎日昼休みに共に筋力トレーニングを行った池田くん，トマトに水やりをしてくれた西保くん，博士課程に進学される笹谷くん，誠に有難うございました．

また，川原研究室秘書の藤田さん，金井さん，水野さんには PFC の部品購入時など非常に多く助けていただきました．特に藤田さんはレタスの育苗作業伴う，トレーの掃除や種まきなどを手伝って頂き，非常に助かりました．

東京大学工作室の内田さんには PFC の組み立てにおける様々な素材加工にあたり，多くの工作機械の使い方を教えていただいたり，実際に加工していただいたりと大変お世話になりました．内田さんのご協力なしには PFC の完成はなかったことと思われます．植物工場におけるデータ解析の研究を 20 年以上前から行っていた東京大学の平藤先生には，直接ご相談し，研究をすすめる上で重要な話を多くいただきました．

グローバル・クリエイティブリーダー育成プログラム (GCL) では共同研究プロジェクト Akiyagri や国際学会の参加など大変貴重な機会を頂き，自身の成長に繋がったことに感謝申し上げます．

最後に，一年を通して身体面・精神面での健康を保って研究活動に専念できたのは両親および彼女のサポートのおかげでした．ここに感謝の意を述べたいと思います．

参考文献

[1] M.W. vanIersel, G. Weaver, M.T. Martin, R.S. Ferrarezi, E. Mattos, and M. Haidekker, “A chlorophyll fluorescence-based biofeedback system to control photosynthetic lighting in controlled environment agriculture,” Journal of the American Society for Horticultural Science, vol.141, no.2, pp.169–176, 2016.

[2] M. Hirafuji and T. Kubota, “Chaos of plant growth under changing environment,” Environment Control in Biology, vol.32, no.1, pp.31–39, 1994.

[3] T. Wakahara and S. Mikami, “Adaptive nutrient water supply control of plant factory system by reinforcement learning,” SCIS & ISIS SCIS & ISIS 2010Japan Society for Fuzzy Theory and Intelligent Informatics, pp.1020–1025 2010.

[4] 農林水産省，“農商工連携研究会 「植物工場ワーキンググループ報告書」,” 2009.

[5] 井熊均，三輪泰史，植物工場経営，日刊工業，2014.

[6] 土屋和，“植物工場をめぐる現状と課題,” 野菜情報= Vegetable information，vol.149，pp.34–44，2016.

[7] Ltd. 株式会社スプレッド SPREAD Co.，“事業戦略,” `http://spread.co.jp/strategy/`，2017. [Online; accessed 31-January-2018].

[8] N. De Souza, “High-throughput phenotyping,” Nature Methods, vol.7, no.1, p.36, 2009.

[9] Y.-K. Oh, B.O. Palsson, S.M. Park, C.H. Schilling, and R. Mahadevan, “Genome-scale reconstruction of metabolic network in bacillus subtilis based on high-throughput phenotyping and gene essentiality data,” Journal of Biological Chemistry, vol.282, no.39, pp.28791–28799, 2007.

[10] J.L. Araus and J.E. Cairns, “Field high-throughput phenotyping: the new crop breeding frontier,” Trends in plant science, vol.19, no.1, pp.52–61, 2014.

[11] A. Udink tenCate, G. Bot, and J. Van Dixhoorn, “Computer control of greenhouse climates,” Symposium on Potential Productivity in Protected Cultivation 87, pp.265–272, 1978.

[12] Y. Hashimoto, “Computer control of short term plant growth by monitoring leaf temperature,” Symposium on Computers in Greenhouse Climate Control 106, pp.139–146, 1979.

[13] Y. Hashimoto, “Recent strategies of optimal growth regulation by the speaking plant concept,” International Symposium on Growth and Yield Control in Vegetable Production 260, pp.115–122, 1989.

[14] 竹内智晴，馬場昭宏，永嶋規充他，“植物工場におけるルッコラ栽培の背丈経日変化に関する多変量解析,” 第 77 回全国大会講演論文集，vol.2015，no.1，pp.517–518，2015.

[15] P.P. Gallego, J. Gago, and M. Landín, “Artificial neural networks technology to model and predict plant biology process,” Artificial Neural Networks-Methodological Advances and Biomedical Applications, pp.197–216, InTech, 2011.

[16] D. Yumeina and T. Morimoto, “Dynamic optimization of solution nutrient concentration to promote the initial growth of tomato plants in hydroponics,” Environmental Control in Biology, vol.52, no.2, pp.87–94, 2014.

[17] D. Yumeina, G. Aji, and T. Morimoto, “Dynamic optimization of water temperature for maximizing leaf water content of tomato in hydroponics using an intelligent control technique,” V International Symposium on Applications of Modelling as an Innovative Technology in the Horticultural Supply Chain 1154, pp.55–64, 2015.

[18] T. Morimoto and Y. Hashimoto, “An intelligent control for greenhouse automation, oriented by the concepts of spa and sfa—an application to a post-harvest process,” Computers and electronics in agriculture，vol.29，no.1，pp.3–20，2000.

[19] T. Morimoto and Y. Hashimoto, “A decision and control system mimicking a skilled grower’s thinking process for dynamic optimization of the storage environment,” Environment Control in Biology, vol.41, no.3, pp.221–234, 2003.

[20] K. Osama, B. Mishra, and P. Somvanshi, “Machine learning techniques in plant biology,” PlantOmics: The Omics of Plant Science, eds. by D. Barh, M. Khan, and E. Davies, pp.731–754, Springer, New Delhi, 2015.

[21] L. Pathak, V. Singh, R. Niwas, K. Osama, S. Khan, S. Haque, C. Tripathi, and B. Mishra, “Artificial intelligence versus statistical modeling and optimization of cholesterol oxidase production by using streptomyces sp.,” PloS one, vol.10, no.9, p.e0137268, 2015.

[22] K. Osama, P. Somvanshi, A.K. Pandey, and B.N. Mishra, “Modelling of nutrient mist reactor for hairy root growth using artificial neural network,” European Journal of Scientific Research, vol.97, no.4, pp.516–526, 2013.

[23] E. Leal-Enríquez and M. Bonilla-Estrada, “Modelling the greenhouse lettuce crop by means of the daily interaction of two independent models,” Decision and Control (CDC), 2010 49th IEEE Conference onIEEE, pp.4667–4672 2010.

[24] K.M. Al-Aubidy, M.M. Ali, A.M. Derbas, and A.W. Al-Mutairi, “Real-time monitoring and intelligent control for greenhouses based on wireless sensor network,” Systems, Signals & Devices (SSD), 2014 11th International Multi-Conference onIEEE, pp.1–7 2014.

[25] R. Salazar, U. Schmidt, C. Huber, A. Rojano, I. Lopez, et al., “Neural networks models for temperature and co2 control.,” International Journal of Agricultural Research, vol.5, no.4, pp.191–200, 2010.

[26] F. Lafont, J.-F. Balmat, N. Pessel, and M. Fliess, “A model-free control strategy for an experimental greenhouse with an application to fault accommodation,” Computers and Electronics in Agriculture, vol.110, pp.139–149, 2015.

[27] C. Harper and M. Siller, “Openag: a globally distributed network of food computing,” IEEE Pervasive Computing, vol.14, no.4, pp.24–27, 2015.

[28] “Amazon — lampwin 45w 植物育成 led ランプ 225 球 ハイパワー 3 色 45w 水耕栽培室内栽培 日照不足解消 植物がすくすく育つ 12ヶ月保証付 — 育苗機,” https://www.amazon.co.jp/gp/product/B07458SYZR/. Accessed: 2018-01-18.

[29] “Rigid hvac - cooling specialist. condensing units — miniature compressors — refrigeration parts,” https://www.rigidhvac.com/. Accessed: 2018-01-19.

[30] “Apache couchdb,” http://couchdb.apache.org/. Accessed: 2018-01-19.

[31] “An open source ecosystem for iot development platformio,” http://platformio.org/. Accessed: 2018-01-19.

[32] “Amazon — buffalo 200 万画素 web カメラ 広角 120° マイク内蔵 ブラック bsw200mbk — バッファロー — パソコン・周辺機器 通販,” https://www.amazon.co.jp/dp/B01LXYHF71/ref=pe_492632_159100282_TE_item. Accessed: 2018-01-19.

[33] “ベビーリーフレタス　スベン【スピーディベジタブル】| 商品情報いろいろ検索 | タネ・苗・園芸用品・農業用資材の総合案内：サカタのタネ,” http://www.sakataseed.co.jp/product/search/code00970902.html. Accessed: 2018-01-19.

[34] J. Snoek, H. Larochelle, and R.P. Adams, “Practical bayesian optimization of machine learning algorithms,” Advances in neural information processing systems, pp.2951–2959, 2012.

発表文献

国際研究会

[1] T. Iizuka, Y. Narusue, Y. Kawahara, and T. Asami, “A Planning Simulation Tool for Energy Harvesting Applications,” Adjunct Proc. 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services (MUSICAL 2016), pp.177-182, Hiroshima, Japan, Nov. 2016.

国際デモ

[2] T. Iizuka, Y. Narusue, Y. Kawahara, and T. Asami, “Planning Simulation Tool for Designing Energy Harvesting Applications,” Adjunct Proc. the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp 2016) , pp.289-292, Heidelberg, Germany, Sept. 2016.

国内研究会

[3] 飯塚達哉，繁田亮，川原圭博，“完全制御型植物工場のための部分観測マルコフモデルを用いた植物生育モデリング” 信学技報, vol. 117, no. 310, ASN2017-69, pp.41-46, Nov. 2017.

[4] 飯塚達哉，成末義哲，川原圭博，浅見徹，“エネルギーハーベスティング型アプリケーション設計を最適化する電力収支シミュレータ，” マルチメディア，分散，協調とモバイル (DICOMO2016) シンポジウム，5E-4，pp.1051-1061, July 2016.

国内全国大会

[5] 飯塚達哉，繁田亮，川原圭博，“完全制御型植物工場における最適栽培環境制御則探索のためのベイズ最適化に向けた成長率モデルの検討，” 情報処理学会全国大会，5M-01, March 2018 (To appear).