

2015 年度 学士論文

デジタル制御ミシンの制作を通じた
衣服分野におけるオープンなものづくりの検討

慶應義塾大学 総合政策学部

森川好美

論文要旨

インターネットの普及とデジタルファブリケーションの発展を背景にして、モノの設計図やデータ、電子回路といった「物質の情報」を共有する文化が生まれている。それらの思想は通常、オープンソースやオープンデザインといった言葉で表されているが、なかでも家庭用 3D プリンタは、ユーザー・コミュニティを中心とした配布と改変の連鎖によって多様な進化をたどっている。

このように、他者との協同によってひとつのモノを開発する機運が生じている一方で、こと裁縫や手芸の分野においては、シェアによるものづくりが普及しているとはいえない。その背景としては、共有を前提としたデジタル工作機械が存在しなかったことや、実空間のローカル・コミュニティが技術と知識の共有の場として重んじられてきたことによる、ウェブ上のコミュニティ＝プラットフォームの不在があげられる。しかし、共創によるものづくりは、裁縫・手芸分野においても変化をもたらす可能性に満ちていると考えられる。そこで本研究では、裁縫・手芸分野におけるデジタル工作機械としてのデジタル制御ミシンの開発を行い、オープンデザインの実践を通して、衣服分野におけるオープンなものづくりの可能性を示す。

キーワード:

デジタルファブリケーション, オープンソース・ハードウェア, デジタル・クラフト

目次

1. 研究背景	4
1.1. 3D プリンタに見るオープンソース・ハードウェアの潮流	4
1.2. ものづくりのための Web コミュニティ	8
1.2.1 Thingiverse	9
1.2.2 Instructables, Fable	10
1.2.3 オープンデザイン	11
1.2.4 ライセンス	11
1.3. 衣服作りを取り巻く環境の変化	12
1.4. デジタルファブリケーション以降の服づくりの事例	14
1.5. 制約が生む創意工夫	16
2. 研究目的	20
3. 従来のコンピュータ刺繍ミシン	21
3.1 問題点	21
3.2 ミシンの転用	22
3.2.1 点字	23
3.2.2 糸による立体的構造	23
3.2.3 自動縫製	24
3.2.4. ジョイント	25
4. デジタル制御ミシンの設計・開発	26
4.1. ミシンの機構	27
4.2. 実装	29
4.2.1. ミシン針との連動	32
4.2.2. プログラム	35
4.2.3. 布を固定する枠	38
5. 実験・評価	39
5.1. デジタル制御ミシンの動作実験	39
5.2. 関連プロジェクトとの比較	40
6. 結論・展望	43
6.1. オープンな機械から生まれる新しいものの作り方	43
6.2. 展望	44
7. 謝辞	45
8. 参考文献	46
9. 図表引用元	48

1. 研究背景

本研究の背景は、大別してふたつの軸からなる。ひとつめは、3D プリンタを代表とするデジタル工作機械を取り巻くオープンデザインという思想を、ハードウェア／ソフトウェア、プラットフォーム、クリエイティブ・コモンズ・ライセンスといった視点から読み解くものである。ふたつめは、日本における洋裁文化の変遷を素描したのちに、デジタルファブリケーションを接続することによる服づくりの未来を展望するものである。

1.1. 3D プリンタに見るオープンソース・ハードウェアの潮流

3D プリンタといったデジタル工作機械に代表されるデジタルファブリケーションは、ここ数年間で急速な進歩を遂げるとともに、一般にも広く知られることになった。その中でもひととき注目を集めている家庭用 3D プリンタの発展に、極めて重要な役割を果たしたのが「Reprap」というプロジェクトである。Reprap の 3D プリンタは、みずからを構成するパーツの約 50%をプリントすることができる「自己複製可能な機械」である[1]。

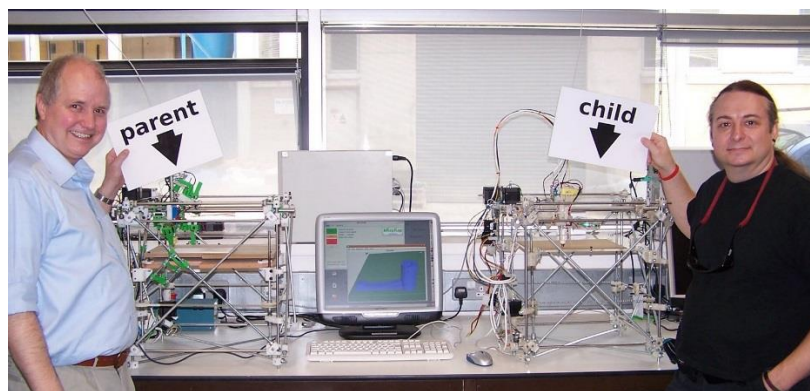


Fig.1 Reprap による自己複製の最初の事例

Reprap をつくるために必要なデータやハウツーはウェブ上で公開されており、誰でも自由に複製、改変、配布することが可能である。そのため、同プロジェクトはインターネットを通じて世界各地へと広がり、多くのメイカーの手によって改良が重ねら

れている。実際のところ、その用途は単なる複製にとどまらず、個性に富んだ派生形が生み出されるにいたっている[Fig.2]。この派生の樹形図は、進化論さながら生物が環境に適応して分岐していく様を思い起こさせる。

その派生の方向としては、主にふたつのベクトルが考えられる。ひとつには、既存の性能を向上させたり、使い勝手を良くすることによる漸進的な派生である。もうひとつは、その素材を PLA（ポリ乳酸）樹脂ではなくペースト状の食品やコンクリートに応用したり、出力サイズを変更することによる抜本的な派生である。これらふたつの派生を臨機応変に選択することで、同じ Reprap を広汎な目的に対応させたり、特定の目的に特化させたりすることができる。こうした自由度が担保されているのも、Reprap の“遊び”をもった設計ならではといえる。[2]

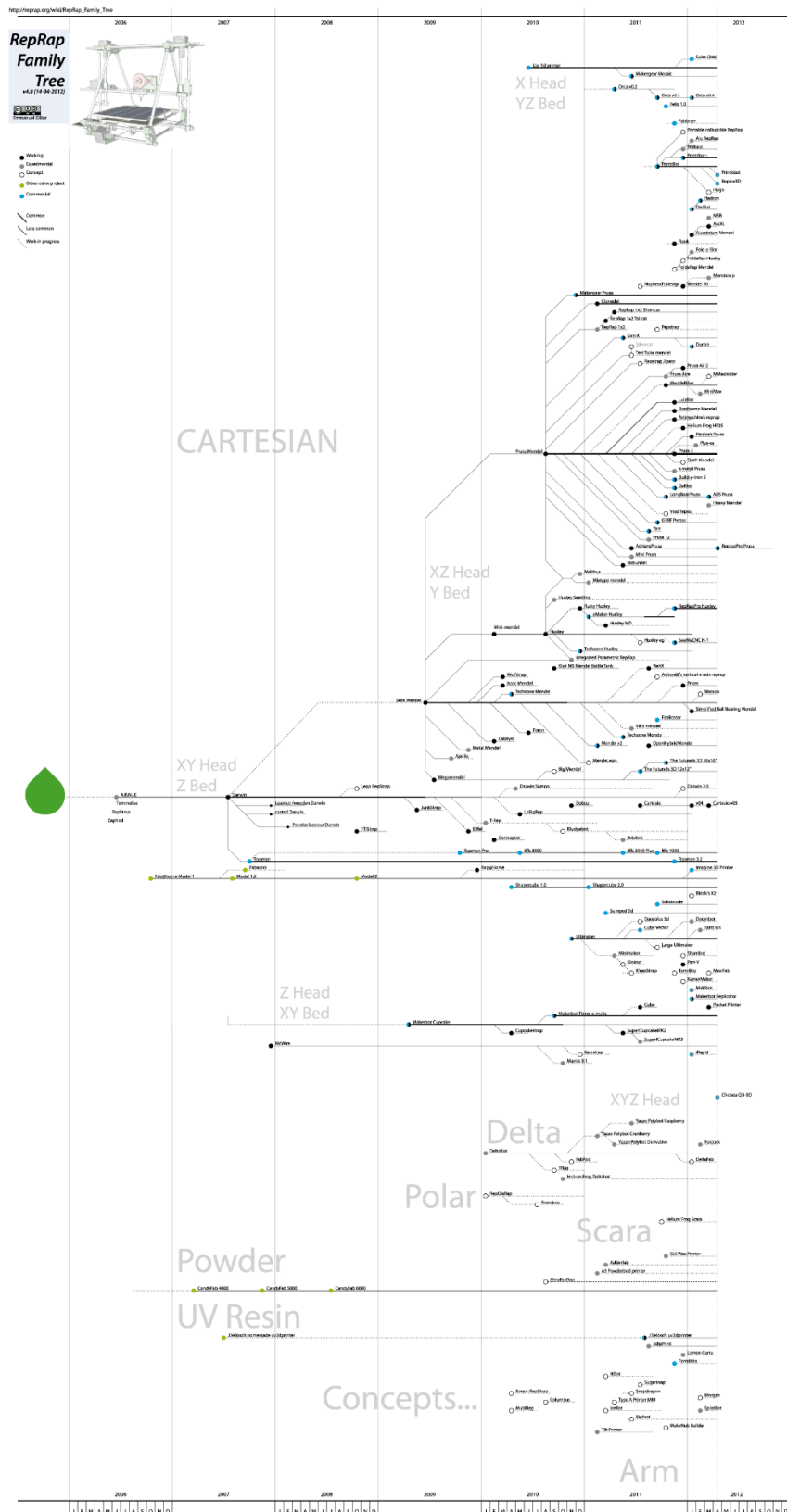


Fig.2 Reprap ファミリーの系統樹

このように、モノの作り方の全部ないしは一部を公開することで他者をデザインプロセスに参画させるという運動は、家庭用 3D プリンタに限らず存在している。その具体例としては、プロトタイプ用の基板である Arduino[Fig.3]のようなプロジェクトがあげられる。これらはオープンソース・ハードウェアと通称され、もともとはソフトウェアの世界において一般的であったオープンソースの思想を取り入れたハードウェアである。

オープンソース・ハードウェアの潮流のもと公開されるデータには、ハードウェア情報（3D データ、回路データなど）とソフトウェア情報が含まれている。ハードとソフトの両面から情報をカバーすることで、誰もが再現かつ改変可能となり、個人のニーズを反映させることができる。いわば、ハードウェアはソフトウェアと手を組むようにしてオープンソースの思想を取り入れようとしているのである。

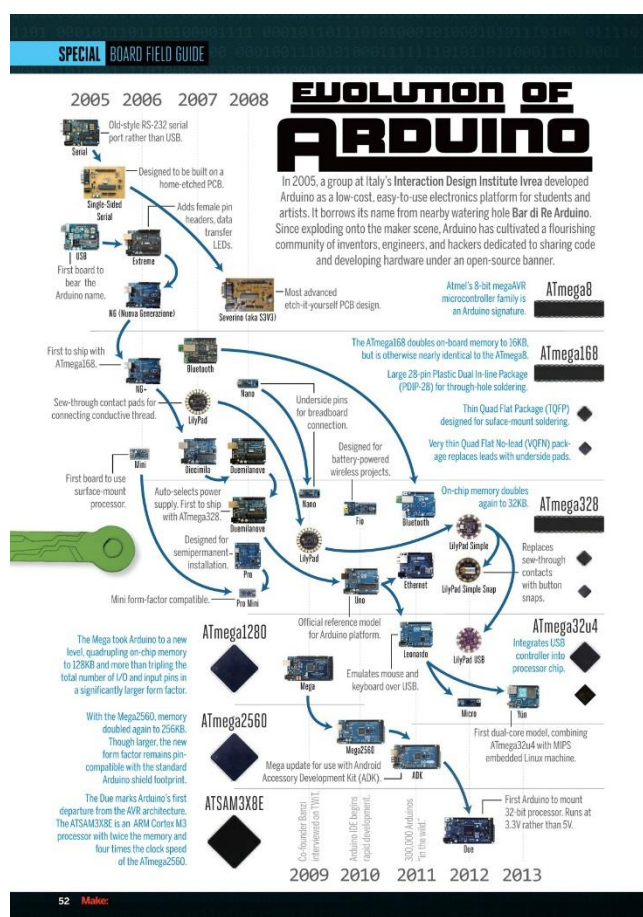


Fig.3 Arduino における系統樹

したがって、Reprap の発展に関しても、付随するソフトウェアの動向を説明しなけ

ればならない。

まず、3D プリンタを動かすには、大別して3種類のソフトウェアが必要になる。具体的に要求されるのは、3D プリンタが命令を受け取るためのファームウェア、3D プリンタに命令を送るためのホストソフトウェア、3D モデルから3D プリンタの制御コードに変換するためのスライサソフトウェアである。しかし、この分類はソフトウェアが合計で三種類であることを意味しない。例えば、ファームウェアに限っても、Reprap.org に12種類のそれが掲載されている[3]。また、これらのソフトウェアもオープンソースの思想によって運用されていることから、ハードウェア同様、多様性に富んでいる。

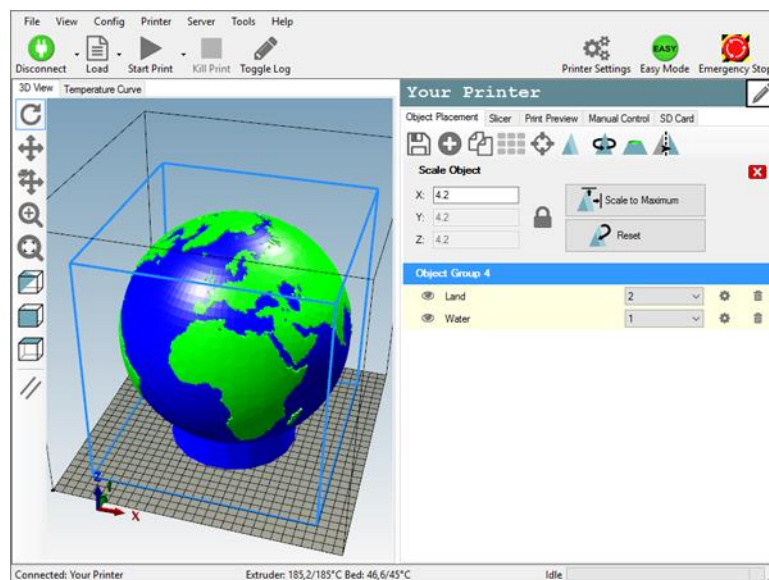


Fig.4 ホストソフトウェアの一つである Repetier-Host

1.2. ものづくりのための Web コミュニティ

Reprap の分析を事例に、ハードウェアとソフトウェアを横断するオープンソースの思想を説明してきた。続いては、オープンソースの運用の場として位置づけられるプラットフォームに関して説明する。具体的には、モノの作り方を共有するためのサービスの事例を通じて、ウェブ上のコミュニティを通じてものづくりに起こっている変化を分析する。

1.2.1 Thingiverse

Thingiverse とは、アメリカの 3D プリンタメーカーである Makerbot が運用するデザイン共有サービスないしはプラットフォームである[Fig.5]。ここには世界各地からアップロードした 3D データが公開されており、ユーザーはウェブサイト上の一覧から好みのデータをダウンロードすることで、そのライセンスのグラデーションに応じて複製、改変、再配布することができる。また、自分がデザインした 3D データをアップロードしたり、データごとに用意されたコメント機能を活用することで、他のユーザーとの交流を行うこともできる。

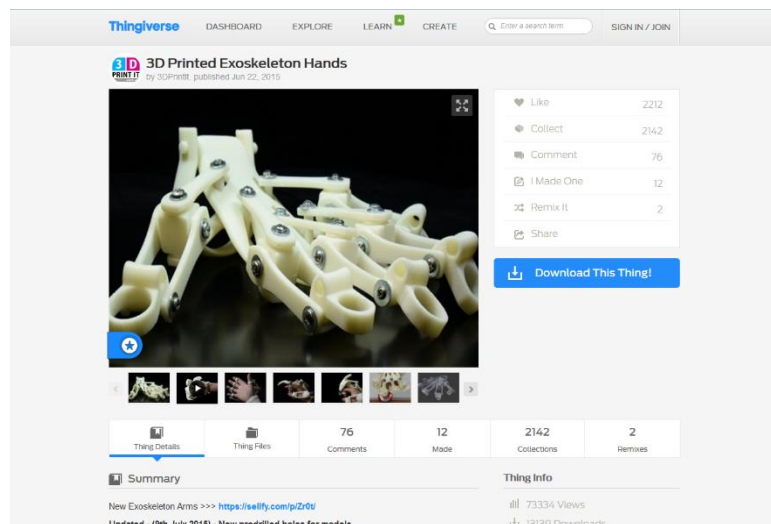


Fig.5 Thingiverse

Thingiverse にはコメント機能の他に Like といった機能が存在するが、なかでもデータの派生プロセスを記録する「Remix」という機能に注目したい。Remix は、Thingiverse 上で公開したデザインを、その派生元である元デザインに関連づけながら公開するための機能である。この機能を使うことで、アップロード側は既存のデータに敬意を払いつつ、改良したデータを公開することができる。また、ダウンロード側からしてみれば、どのような派生を経たのかを追って参照できることから、アップロード側、ダウンロード側を問わずメリットを享受することができる。このようにデータの公開と派生を活性化させる仕組みは、Thingiverse におけるオープンソースの促進に一役買っているといえるだろう。

1.2.2 Instructables, Fable

Thingiverse の他にも、Instructables や Fable というプラットフォームが存在している。

Instructables のアーキテクチャは、モノの作り方の共有に特化したつくりになっており、DIY 家具から料理、コスプレといった幅広い分野のレシピが公開されている。ユーザーは独自のものづくりレシピを公開するのみならず、レシピを参考に作ってみた感想をコメントや「I made it」、「Favorite」という機能を通じてフィードバックとコミュニケーションを行うことができる。

Fable もモノの作り方を共有するサービスであるが、レシピの多様性に富んだ Instructables とは異なり、そのカテゴリーはデジタルファブリケーションにフォーカスされている。

実際のところ、Fable にはオープンソースの思想のもとプロジェクトを運用するための機能が備わっている。具体的には、完成したレシピを公開する他にも、プロジェクトに関するドキュメントを複数のユーザーで編集するための機能などが存在する。その進行中、進行外を問わずにプロジェクトが並列されて公開されている様子からも、Fable が協同的なものづくりを推進していることがうかがえる。

このサービスの原型になっているのは、ソフトウェア開発のための共有サービスである Github である。このサービスの隆盛からもうかがえるように、ソフトウェアの分野においては、オープンな開発が当然のように行われている。というのは、オープンソースの思想のもとプロジェクトを公開すれば、無数のステークホルダーによって随時バグレポート、レビュー、アドバイスがアップロードされるからである。少なくともソフトウェアにおいては、一人の人間による閉ざされた開発ではなく、無数の人間による開かれた開発が劇的な推進力を発揮するといえる。

以上のように、Instructables と Fable は Thingiverse と同じものづくりのためのプラットフォームである一方で、モノのデータではなく作り方に重点を置いているという特徴がある。とりわけ Fable は、オープンソースの思想をものづくりの世界に輸入しようとする野心的な試みであることが見てとれる。

1.2.3 オープンデザイン

かつてソフトウェアの世界で起こったオープン化の動きは、ものづくりの分野にまで広がっている。そこで、これらの新しいものづくりの動向を包括するオープンデザインという概念について触れることにする。

オープンソースにおいて共有されるものはソースコード、すなわち情報であったが、オープンデザインにおいてはその射程が物質にまで拡張されることになる。具体的には、単なる意匠としてのデザインのみならず、設計図や組み立て方といった、デザインのプロセス全般が共有されることになる。これも、前掲の Instructables のようなプラットフォームの登場によって可能になった現象である。

みずから能動的に他者を巻き込むオープンデザインにおいては、他者の理解の理解としての二次的理解に留意しなければならない。あつかう情報が他者による再現性を保つためには、どの情報をどのように公開し、どのようにして関わらせるのかといったことも含めてデザインする必要があるのである[4]。

1.2.4 ライセンス

以上のように、デジタルファブリケーションをきっかけに、ウェブを介したものづくりコミュニティが発達している。これらのサービスは全てクリエイティブ・コモンズ・ライセンス[Fig.6]を採用している。以下にクリエイティブ・コモンズ・ジャパンによる説明を引用する。

“CC ライセンスとはインターネット時代のための新しい著作権ルールで、作品を公開する作者が「この条件を守れば私の作品を自由に使って構いません。」という意思表示をするためのツールです。

CC ライセンスを利用することで、作者は著作権を保持したまま作品を自由に流通させることができ、受け手はライセンス条件の範囲内で再配布やリミックスなどを行うことができます。[5]”

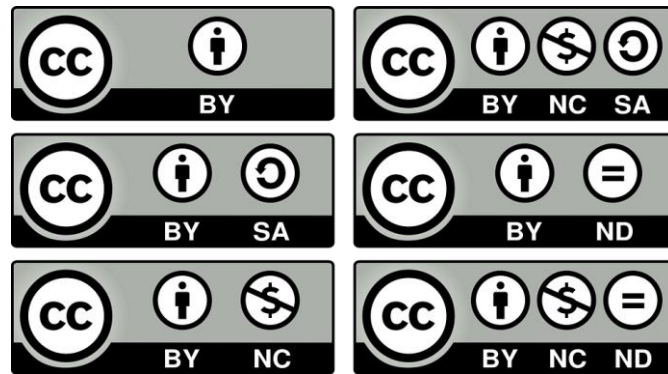


Fig.6 クリエイティブ・コモンズ・ライセンス 4.0 一覧

インターネットの普及によって生産者と消費者の境界が曖昧になるなかで、ライセンスと権利の問題も複雑化し、従来の著作権法ではカバーし切れなくなってしまった。そこで要請されたのがクリエイティブ・コモンズ・ライセンスである。完全なる保護か完全なる放棄というバイナリで記述されていた著作権に、中間のグラデーションを持ち込んだクリエイティブ・コモンズ・ライセンスを用いれば、最低限の権利を守りつつ、複製や改変を奨励することができる。

また、ライセンス利用者が個人であることがままあると想定されることから、ライセンスの内容は専門家以外でも理解しやすいよう工夫されている。具体的には、全6種類のライセンスが4種類の単純な条件の組み合わせで構成されていることや、ライセンスの表明に視覚的なアイコンを用いるという設計を見てとることができる。このように、オープンなものづくりを支えるプラットフォームの整備はライセンス領域にまでおよんでいるといえる。[6]

ハードウェア／ソフトウェア、プラットフォーム、クリエイティブ・コモンズ・ライセンスといった視点から、オープンソースやオープンデザインといった概念を分析してきた。次項以降では、本研究がこの思想を応用しようとしている洋裁文化の歴史について記述する。

1.3. 衣服作りを取り巻く環境の変化

第一次産業革命が織機と紡績機の発明から始まったように、衣服の生産は人間生活の発展に欠かせないものであった。現在私たちが着用している既製服がこれほど供給されるようになった背景にも、縫製ミシンの普及があげられる。生地のための糸をつく

る紡績機、生地をつくる織機、そして生地を縫製するためのミシン、これらの技術からなる大量生産のインフラが、現在にいたるまでファッションを支え続けているのである。このことは、約 150 年間にわたってミシンの形状に大きな変化がないことから明らかである[7]。



Fig.7 1854 年に製造されたシンガー社製ミシン

19 世紀以前は、世界中のどこにも既製服が存在しなかった。すなわち、注文によって仕立てられた服か中古服以外は、すべて家庭内でつくられていたということになる。つまり、家庭を支える女性が洋裁の主体を担うことで、衣服が供給されていたのである。

しかし、現代日本に焦点を当ててみると、一家に一台と言われるまでに普及したミシンはもはや見る影もなく、かといって手作業による針仕事が復興したわけでもなく、洋裁文化は途絶えてしまったように見える。「生産者」と「消費者」を明確に分別したファスト・ファッションの台頭によって、家庭における個人規模の洋裁文化は淘汰されてしまったのである[8]。

しかし、インターネットの普及とデジタルファブリケーションの発展によって、洋裁が新しいかたちで蘇ろうとしている。その鍵となるのが、すでに触れたデジタルファブリケーションとオープンデザインである。産業革命がファスト・ファッションのインフラを築いたように、デジタルファブリケーションによるものづくり革命が、新しい洋裁のインフラになる可能性があるのである。

1.4. デジタルファブリケーション以降の服づくりの事例

デジタルファブリケーションによるものづくり革命が、新しい洋裁文化をもたらす可能性について言及した。そこで本節では、デジタルファブリケーションを前提とした新しい服づくりの先行事例をみつつほど取り上げてみたい。

まずは、現在進行形で開発されている世界初のデスクトップ型ファブリック 3D プリンタ、「Electroloom」を取り上げたい。この機械は、型紙に沿って裁断した生地を縫い合わせる従来の服づくりによらず、プリンタ内に入れた型に繊維の素となる液体を吹き付けることで衣服を出力するプロセスを提案するものである。プリンタの中に入る型が素材を問わないことから、レーザーカッターや 3D プリンタを組み合わせることで、個人でも簡単に衣服をデザインすることができる。[9]



Fig.8 Electroloom

次に、Fablab Barcerona が中心となって行っている「Fabtextiles」というプロジェクトについて説明する。これは、服づくりにおける従来の手法とデジタルな手法を組み合わせたマテリアル・プロトタイピングを通じて、ファブ前提環境における新しい服づくりの実践を行うものである。その提案は多岐にわたっており、3D スキャナとパラメトリック・モデリングによって個人の身体にフィットする衣服や、レーザーカッターでカットしたジョイントパターンによって生地同士を接合することによる無縫

製の衣服などがあげられている。[10]

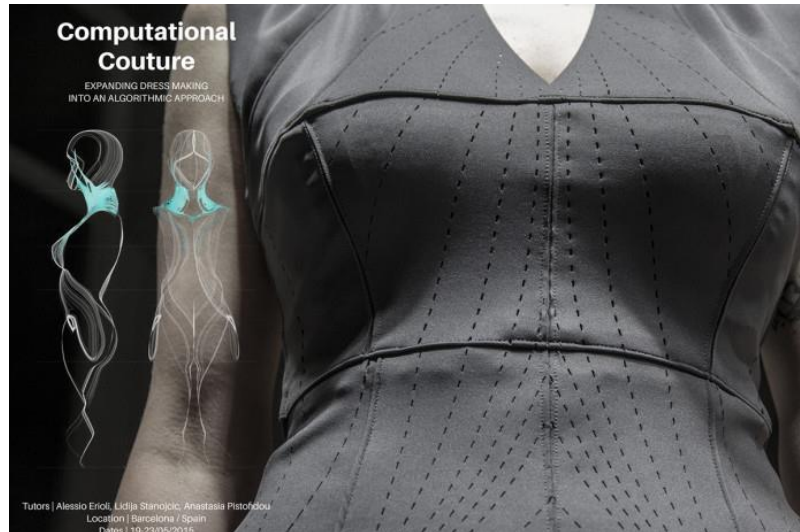


Fig.9 FabTextiles による Computational Couture

最後に、nervous system が提案している 3D プリンタを用いた衣服「Kinematics Dress(図 10)」を取り上げる。これは SLS 方式の 3D プリンタによって出力されたドレスであるが、小さなパーツをヒンジで結合することによって構成されていることから、硬いナイロン素材にも関わらずドレープが生じるという性質を持つ。ドレスのシルエットや動きを画面上でシミュレーションしたうえで、意図した通りの表情を再現できるという点は、従来の方法では実現できない点だろう。また、ヒンジの可動部分を応用し、折り畳まれた状態で出力されることから、造形可能範囲や組み立てといった制約をクリアしている点も見逃せない。[11]



Fig.10 nervous system による Kinematic Dress

以上、デジタルファブリケーションを前提とした新しい服づくりの事例を紹介した。Electroloom は電界紡糸という他分野の技術を転用した機械の開発を通じて、Fabtextile はデジタル工作機械の強みを活かすデザインの実践を通じて、Kinematic Dress は高度な設計／シミュレーションを可能にするソフトウェアによって、従来とは異なる服づくりのプロセスの提案をしている。つまり、これらは新しい技術の衣服への応用方法を模索した潮流であると整理できるだろう。

1.5. 制約が生む創意工夫

デジタル工作機械を用いた新しい服づくりの潮流について確認した。そして、それらはいずれもファッションにおけるテクノロジーの応用可能性を探究するものであった。

しかし、Reprap の節でも記述したように、デジタルファブリケーションの可能性はその技術的な目新しさに留まるものではない。むしろ、デジタル工作機械がオープンソースやオープンデザインといった思想と手を取り合い、プラットフォームで実践されて初めて、継承と派生からなるオープンな生態系が構築されるのである。つまり、デジタルファブリケーションの服づくりへの応用にあたっては、工作機械とそのコミュニティを介した人々の想像力は無視できない。

そこで本節では、筆者自身が課外活動として継続してきた3回のワークショップについてそれぞれ記述する。

まず、慶應義塾大学湘南藤沢メディアセンター内のデジタルファブリケーション工房「Fabspace」にて、デジタル工作機械を用いた初心者向けの手芸ワークショップを実施した。「女子ファブ」と題したこのワークショップであるが、筆者の予想を裏切り、内容を変えて実施された二回のどちらにも男子学生が参加する運びとなった。

最初に行ったワークショップは、自分でデザインしたポーチをつくるという内容であった。使用するデジタル工作機械としてはレーザーカッターと職業用ミシンを設定し、ボタンのデザインには Adobe 社が開発したソフトウェアである Illustrator を使用した。体験の流れとしては、あらかじめ用意された型紙を編集すると平行して、ボタンをデザインし、出力後に統合、完成させるといったものである。主催者側で用意したポーチとボタンの型紙は、至極シンプルなものであった。しかし最終的には、参加

者が手を加えることによって、同じ型紙からデザインしたとは思えないほど個性に富んだ派生作品が生まれるにいたった。例えば、Fig.11 右下の作品は、ボタンで目やくちばしといった顔のパーツを表現することで、ポーチ全体を鳥の顔に見立てたデザインを制作している。素材と時間の制約がありながら、作品にこのような多様性が生まれることは非常に興味深いといえるだろう。



Fig.11 第一回女子ファブで制作されたポーチ

2 回目に実施したワークショップでは、くるみボタンの土台に布をくるむ手法を応用することで、立体的なブローチを制作した。今回使用したデジタル工作機械はデジタル刺繍ミシンと 3D プリンタであり、それぞれ表面の布と土台を出力した。今回もあらかじめ主催者側でモデルを用意し、3D プリンタによる土台は参加者の希望サイズに合わせてリサイズした。参加者にはそのサイズの制約のもと、刺繍データを作成していただいた。参加者のひとりの作品である Fig.12 は、円形の土台の形状を硬貨に見立てた刺繍を施している。



Fig.12 第二回女子ファブで制作された刺繍ブローチ

どちらのワークショップでも、制約のなかで発揮される参加者の独創性が印象的であった。

この後、筆者は素材に重きをおいたワークショップを企画した。具体的には、2015 年 11 月 20 日・21 日に六本木ミッドタウンで開催された「Open Reserch Forum 2015」において、アドビシステムズ株式会社スポンサーのもと、高校生向けデザインのワークショップを担当した。そのタイトルとして掲げられたのは、「身の回りのモノをハックしてデザインしよう」という標語である。その言葉の通り、我々の身の回りにあふれた「完成品」である日用品を、未完成な「素材」として転用することを目的とした。

その具体的な内容は、あらかじめ主催者側でテストした 100 円ショップの日用品、約 10 種類から好きなものを選択し、名刺をつくるというものである[Fig.13]。主催者側であらかじめ選定した素材のなかには、コルクボードやクリアファイルといった比較的加工に用いられやすいものから、アルミバック、携帯灰皿、CD-R と多岐にわたった。

文字の出力は、CMYK による印刷のかわりにレーザーカッターでの彫刻機能で行った。そのため、一方ではコルクボードのように焦げがはっきりと表れたり、他方ではクリアファイルのように溶けて色が付かなかったりと、選定した素材のなかでも明らかな向き不向きが見られた。

しかし、ここにおける向き不向きの基準は普遍的なそれではなく、つくるものによって左右されるといえる。つくるものが決まってはじめてつくる方法が決まり、制約が生じるのである。モノの形状がたやすくデータ化される現代ではあるが、その加工方法や素材に対する視点を忘れてはいけないということを、あらためて実感させられたワークショップであった。

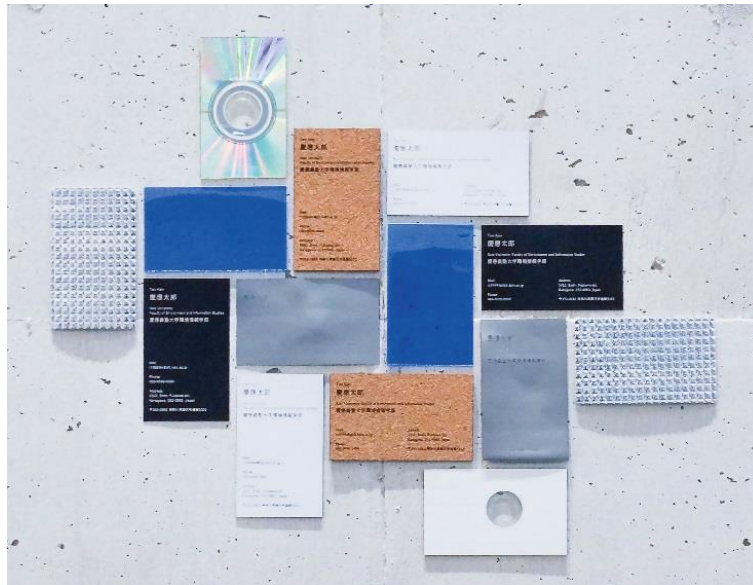


Fig.13 ワークショップのために筆者が制作したサンプル

筆者が主催した3つのワークショップを振り返ることで、デジタルファブリケーションにおける想像力について考察した。強い制約と初心者という条件にも関わらず、いずれの事例においても個性に富んだ作品が生み出されていた。

機械そのものの複製と改変を良しとするデジタルファブリケーションのものづくりにおいては、(ほぼ)あらゆるものをつくることができる。しかし、なんでもつくれるという無限の可能性そのものが創造に貢献することはない。むしろ、元ネタとなるデータが存在することで多様な派生が生み出されることは、Reprapの3Dプリンタの事例からも明らかである。

つまり、デジタルファブリケーションにおいては、元となるデータや制約が存在し、思考のベクトルを限定してはじめて、人々の想像力がかきたてられるのではないだろうか。これがワークショップを通じた筆者の感想である。

2. 研究目的

オープンソース、オープンデザインの数々の実践が明らかにしていることは、モノの作り方の全部ないしは一部の公開によって他者を参画させることは、閉ざされた開発では起こりえない創発的な派生が生じるということであった。その代表的な事例である Reprap の 3D プリンタから、デジタルファブリケーションを前提としたオープンなものづくりの条件を以下のように考えることができる。

- 1,設計が改変可能、かつ公開されている [機械の要件]
- 2,他者にシェアできる [データの要件]
- 3,共有の可能な場がある [環境の要件]

昨今のデジタルファブリケーションの普及にともない、裁縫と手芸の分野においても新しい方法を模索するプロジェクトがいくつか存在するが、以上の条件を満たすデジタル工作機械はいまだ存在していない。なまじ歴史があり、伝統的な方法が重んじられる当分野において変化を起こすには、自由な改変を歓迎する機械の開発が必要条件である。

そこで本研究では、かつては家庭にも普及し、ありふれていた機械であるミシンに着目する。ミシンは一見用途が限定された制約の多い機械である。しかし、その制約が逆に人々の創造性を引き出す可能性は前章末節で述べた通りである。

以上のことを踏まえ、本研究では裁縫、手芸の分野におけるオープンなものづくりを実現させるための開発を行う。具体的には、誰にでも制作することができるデジタル制御ミシンの作り方の提案を通じて、今後の裁縫、手芸の分野における草の根的発展を促進する契機を生み出すことを目的とする。

3. 従来のコンピュータ刺繍マシン

デジタル化がなされており、一般にも流通している既存のマシンとしては、コンピュータ刺繍マシンの存在があげられる。これはパッケージとして購入、あるいは自分で制作したデータをもとに、刺繍枠に取り付けられた生地に刺繍を施す機械である。コンピュータ刺繍マシンはマシンメーカー各社から販売されているものの、通常のマシンの高級機として位置づけられていることから、あまり普及していないという現状がある。また、後述するように純粋に制作を阻害する機能的な制約も多い。

そこで本章では、このコンピュータ刺繍マシンが抱える問題の実践を通じて分析していく。なお、コンピュータ刺繍マシンには家庭用と業務用という区分が存在するが、本章で取り扱うのは家庭用のそれとする。

3.1 問題点

一般的なコンピュータ刺繍マシンは、通常直線縫いマシンに刺繍機を装着し、縫い方向をコンピュータ制御することで刺繍する。その刺繍機に生地を固定した刺繍枠を取り付け、針の上下運動に同期させながらXY軸方向に制御することで、任意の模様を刺繍することができる。縫う速度はおおよそ400rpmから1200rpmほどで、名入れ刺繍程度であれば数分で完成する。特徴としては、手刺繍では難しく時間がかかる図柄でも、短時間で正確に刺繍することができる[Fig.14]。その一方で、刺繍の領域が刺繍枠の範囲に限定されていることから、加工可能なサイズや素材が限られてしまう。例えば、刺繍枠より小さな生地や、挟み込んだときにシワがついてしまう紙のような素材には刺繍をすることができない。



Fig.14 コンピュータ刺繍マシンによる刺繍作品

データ作成の手順は、次の通りである。図柄の元となる画像ファイル(主に jpg、png などのラスター形式、または ai、wmf などのベクター形式)を制作し (CAD)、専用ソフトウェアを用いて刺繍データに変換する (CAM) という二段階の工程が必要になる。

写真のように色数が多い画像の場合は、変換の段階で減色処理を行う。また、ソフトウェアによっては縫い方の種類や縫い目の幅、角度などを調整する機能が備わっているものもある。コンピュータ刺繍マシンのデータ形式はメーカーごとに異なり、互換性がないことから、同じデータを使い回すには変換前の画像ファイルを共有しなければならない。また、扱える画像ファイルの形式もソフトウェアによって異なるために、3D プリンタにおける.stl のような拡張子のデファクト・スタンダードが存在していない。

言わずもがな、複製と改変が容易なデジタルの特徴を活かすには、データは共有しやすい形式であるべきである。この煩雑さがコンピュータ刺繍マシンにおけるデータ共有やコミュニティ生成の障害に拍車をかけているといえる。

以上のような制約を前提に、あくまでコンピュータ刺繍マシンの可能性を引き出すべく4つの実験を行った。次節以降ではその経過を述べていく。

3.2 マシンの転用

コンピュータ「刺繍」マシンは、その名称から想起される機能やプリセットされているデータから、名入れやキャラクターのアップリケ作りといった装飾に使われること

が一般的である。

しかし、コンピュータ刺繍ミシンをデジタル工作機械のひとつであるから見立てると、もっと別の使い方があるのではないかと考えられる。そこで筆者は、刺繍を「布地上で糸と糸を絡み合わせることで固定すること」と再解釈し、刺繍という枠組みをこえるための 4 つの実践を試みた。なお、使用したコンピュータ刺繍ミシンは Innovis D400J (Brother) と Memory Craft (JANOME) である。

3.2.1 点字

まず、Fig.15 のようにコンピュータ刺繍ミシンによる点字の刺繍をおこなった。

制服への名入れをはじめとして、文字をモチーフにした刺繍は多い。今でこそ高精度なファブリックプリンタが開発され、布に文字を印刷することができるようになったが、以前は布に対して文字を刻むことは難しかった。そういった意味で、刺繍は布の上に複雑な模様をつくるもっとも原始的な方法のひとつである。

しかし、布に文字を印刷する方法が多様化した現代においては、刺繍という手法の利点を問い直す必要があるのではないだろうか。刺繍は印刷とは異なり、物質としての糸そのものを織り込むことで布を加工する。だとすれば、この物質性に着目することで、皮膚感覚に訴えるものをつくる方法として再解釈できるのではないだろうか。したがって、視覚的な文字でありながら触覚的な質感をあわせもつ点字は、この問題意識に最適なアプリケーションのひとつであるといえる。



Fig.15 刺繍による点字

3.2.2 糸による立体的構造

刺繍の試行錯誤を行うなかで、糸密度を高くすると、糸同士が絡み合って硬くなると

同時に柔らかさをあわせ持つことを発見した。この特性を流用して、糸だけで構造を設計するべく Fig.16 のプロトタイピングを実施した。

コンピュータ刺繍ミシンには、水に溶ける水溶性シートに刺繍を施し、刺繍後にシートを溶かすことでレースのようなものをつくる小技が存在する。その水溶性シートを用いて、糸だけで柔らかな構造をつくる実験を行った。具体的には、立体の展開図となるデータを作成し、水溶性シートを溶かした刺繍を手縫いで組み合わせることで、糸のみで立体を成形することができた。おそらく、単体では細くて柔らかい糸が密に絡み合うことで張力が発生し、全体を安定させていると推測される。また、本来は平面的な刺繍であるが、組み合わせることで高次の立体を成形できることを発見した。この組み合わせを機械の手に委ねることで、3D プリンティングのように出力できるようになったとしたら、立体的な糸の構造を設計するための CAD ソフトウェアが必要になってくるだろう。

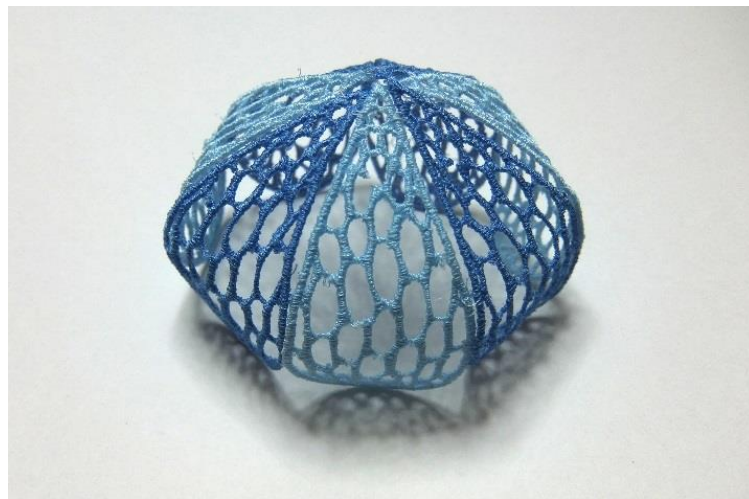


Fig.16 刺繍による立体形状

3.2.3 自動縫製

コンピュータ刺繍ミシンは加工を開始すると、糸が切れるといった不測の事態が起こらないかぎり自動で縫い続ける。この利点から、データをもとに自動縫製を行う機械として転用できるのではないかと考えて、Fig.17 の実験を実施した。これは、重ね合わせた布を縫い合わせのデータに沿って縫うことで、通常の縫製作業をデジタル化するという試みである。結果として、立体的な衣服のようなものを縫製するためには、布地の間に綿やウレタンフォームといった緩衝材を挟む作業が必要であることがわかった。



Fig.17 自動縫製の実験

3.2.4. ジョイント

従来の服づくりは、ニットを除き生地を縫い合わせることで衣服の形状をつくっていたが、ミシンを扱えるようになるには技術の習得が必要である。だが、ミシンの習得は容易ではないことが、洋裁のハードルを上げている。そこで、ミシンで縫い合わせる必要のない服づくりの方法を検討するために行ったのが Fig.18 の刺繍によるジョイントの試作である。Fig.16 の作例では密度の高い刺繍がある程度の硬さをもつことを立体的な構造に応用したが、それと同様に硬さをジョイントに応用すれば、縫い合わせるのではなく、ボタンのように留めることで衣服をつくる方法を実現することが可能になるだろう。



Fig.18 刺繍によるジョイント

4. デジタル制御ミシンの設計・開発

コンピュータ刺繍ミシンはデジタルデータによって制御されているものの、他のデジタル工作機械とは異なる発展をたどってきた。

コンピュータ刺繍ミシンは、従来のミシンの機構はすえおきに、XY 方向に稼働する刺繍枠と専用のソフトウェアから書き出されたデータによって制御される。用途としては、一般家庭での利用や商業利用を前提とした製品であることから、ハードウェア、ソフトウェアともに動作の安定が追求されており、想定外の使用は制限されている。しかし、ミシンの「糸によって布と布を縫い合わせる」という根源的な機能がデジタルデータと組み合わせられることによる可能性は、単なる刺繍に留まらないことを前章で示した。

Reprap をはじめとする家庭用 3D プリンタが、オープンな開発によって思わぬ派生をとげたように、ミシンも開発者の想像を超えた派生の可能性を秘めている。そのためには、クローズドに開発されてきたミシンを専門家でなくても扱えるようにする必要がある。そのためのケーススタディとして、本研究では G-code によって制御できるデジタルミシンを実装し、その作り方を公開する。この Reprap にならったオープンソース・ハードウェアの実践を通じて、ミシンの生態的発展の契機を生み出すことを目的とする。

実装の具体的な内容は別してふたつあげられる。

ひとつめは、G-code 制御によるデジタルミシンの実装である。従来のコンピュータ刺繍ミシンを参考に、XY 方向に稼働する枠に生地を張り、ミシンの針の下に設置した。これによって、G-code の命令に通じて任意の形状を縫うことができる。

ふたつめは、デジタルミシン専用の G-code 生成ソフトウェアの制作である。ビジュアルデザイン言語の Processing を通じて、上記のデジタルミシンを制御する G-code を吐き出すソフトウェアを開発した。この G-code の仕様は、一般的に 3D プリンタの制御に用いられている G-code をベースに、独自の文法を定義したものである。

4.1. ミシンの機構

ミシンの種類は家庭用、職業用、業務用と様々であるが、本研究では主に直線縫いをするミシンのなかでも家庭用のものを採用した。その家庭用ミシンも、制御方式の違いによってコンピュータミシン、電子ミシン、電動ミシンの 3 種類に分類される [Fig.19]。

まず、コンピュータミシンは内蔵されたコンピュータの制御によって、プログラムに応じた多様な縫い方をすることができる。なお、一般的な刺繍ミシンはコンピュータミシンをベースに刺繍機能を追加したものである。

電子ミシンは電子回路を持ち、電子制御によって針の上下や速度を制御している。速度は段階化されているものの、一定である。

最後に、電動ミシンは電子制御のための回路を持たず、フットコントローラーによってモータの回転を操作するほかは機械によって動作する。速度はアナログで変化することから不安定である。

本制作においては、仕組みがもっとも単純かつ安価で手に入りやすい電動ミシンの使用が適切であると考えた。

	制御	針の停止位置
電動ミシン	なし	不定
電子ミシン	電子基板	一定
コンピュータミシン	コンピュータ基板	一定

Fig.19 家庭用ミシンの種類

ミシンの基本機能である上糸と下糸の縫い合わせ [Fig.20] であるが、制御方式による機構の違いはほとんどないと言ってよい [Fig.21]。まず、機械の右下に取り付けられたメインモータの動力を上軸に伝え、針棒クランクとクランクロッドによって針棒の上下運動へと変換する。次に、ミシン針中央の溝に沿って、輪をなした上糸が布地の下の下糸釜の剣先に引っかかる。そして、中釜の回転とともに広がった上糸の輪に下糸が通り、針が戻ることで結び目ができる。以上がミシン縫いを支える主たる機構である。

その他、針の下の送り歯によって布を手前から奥へ送る機構と、縫い目を整えるために上糸の張りを調整する補助的な機構があり、これら 3 つの機構がミシンの基本的な機能を構成している。本研究で制作するデジタルミシンにおいては、糸による縫い合わせの機能はすえおきに、布を送る機能をデジタル制御で代替することにした。

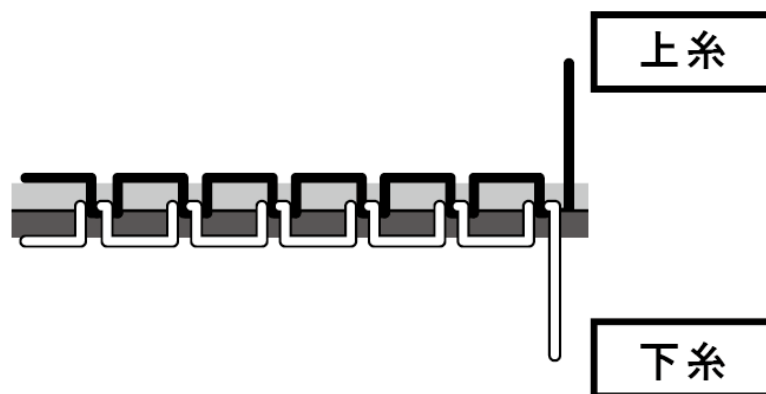


Fig.20 ミシン縫いの仕組み

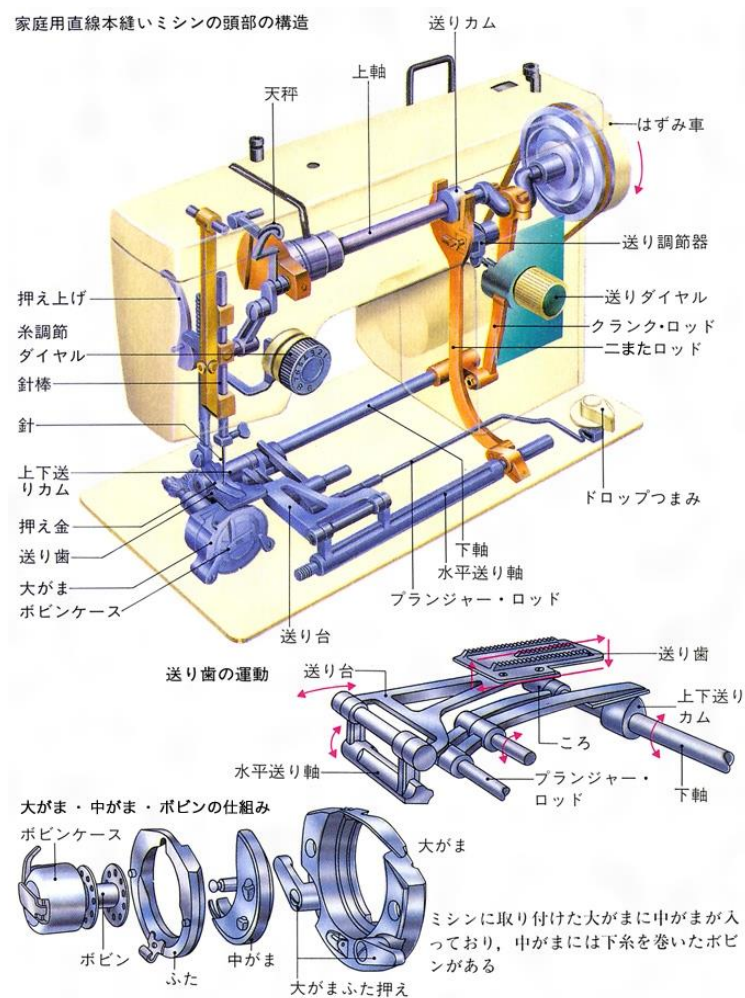


Fig.21 ミシンの基本機構

4.2. 実装

議論をスムーズにするために、本節では本研究におけるミシンのデジタル制御について定義する。

通常のみシン縫いでは、上糸と下糸が絡まった縫い目が形成される。縫い目は基本、送り歯によって直線状に形成されるが、人の手によって布の進向方向を微調整することで、曲線状に縫うことも可能である。その直線、曲線を問わず、一針分の縫い目は一定の長さの線分に分割されている[Fig.22]。

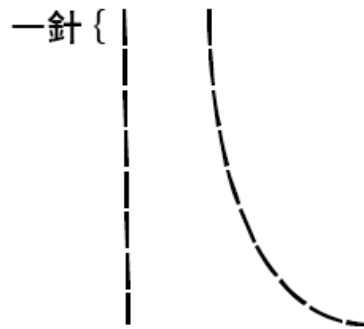


Fig.22 ミシン縫いの縫い目

これに対して、本研究におけるミシンのデジタル制御とは、針を入れる場所を XY 平面上の座標によって決定し、点と点を結んだ線分による縫い目を形成することと定義する [Fig.23]。したがって、縫い目は通常のみシンと異なり一定ではない。線分は始点と終点を持ち、前の線分の終点が次の線分の始点となる。ただし、厳密には糸が太さを持つことから、図一のように、前後の線分が点を共有することはない。

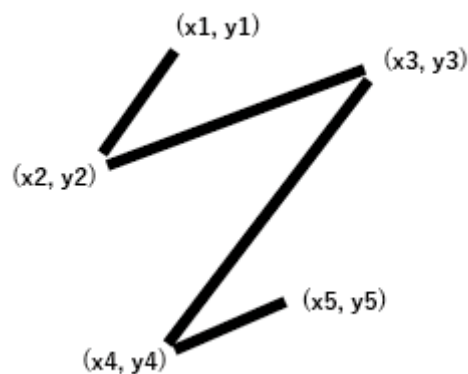


Fig.23 デジタル制御ミシンの縫い目

次に、デジタル制御ミシンの実装方法について詳細を述べる。のちの公開を意識して、初心者でもなるべく手軽につくれる作り方を目指した。そのためには、専門的な機材がなくても、デジタル工作機械によってつくることができる設計が好ましいといえる。とはいえ、ミシンは精密機械であり、個人で一から制作することは難しいことから、既存の機械を組み合わせる改造を加えるという方法を採用することにした。したがって、以下に選定したミシンおよび XY 軸プロッタと、その理由を述べていく。

オープンデザインの事例のひとつに、IKEA で販売されている家具に手を加え、寸法

や外見を変える「IKEA ハック」という文化がある。世界中の「IKEA ハッカー」たちは、ウェブ上に公開されたレシピを参照しながら、世界中のどこでも手に入る IKEA 製の家具をアレンジし、シェアしている。大量生産ならではの規格化と入手可能性が、IKEA ハックが世界的に波及した要因であるといえるだろう。

このような背景を踏まえて、本研究で使用するミシンにはフットコントローラーで動作する IKEA 製の電動ミシンを選択した。なお、本体にはレバーを回すことで針棒に伝動するカムを入れかえ、縫い目に変化をつける機構が備わっているが、本研究では直線縫いのまま使用する。



Fig.24 電動ミシン(IKEA)

XY 軸の駆動機構としては、Makeblock 社の XY Plotter Robot Kit[Fig.25]をベースに改変することにした。

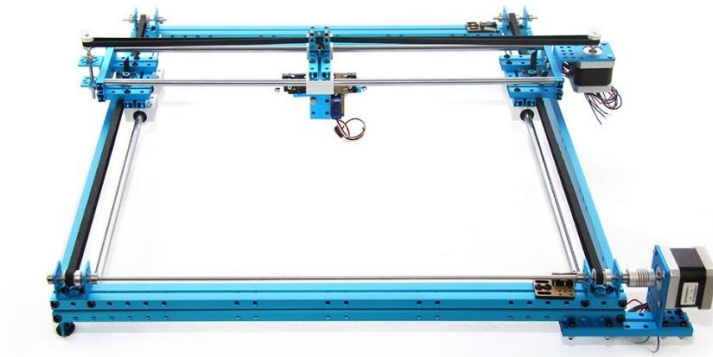


Fig.25 XY Plotter kit (Makeblock)

Makeblock は、LEGO のような統一規格のパーツを組み合わせることでロボットをつくることのできるキットやパーツを製造している。この XY Plotter Kit は、制御ボードに接続されたステッピングモータを動かし、タイミングベルト・プーリーと連動することで、XY 軸に組まれたシャフトのリニアガイドが動くという仕組みになっている。



Fig.26 XY Plotter Kit の部品一覧

制御ボードには Arduino Uno の互換基板が用いられており、Reprap をベースにアレンジされたファームウェアを書き込むことで、G-code による制御を行うことができる。すなわち、Reprap のコミュニティに蓄積されてきたノウハウを流用できるというメリットがあることから、この機種を選定するにいたった。このキットの本来の用途は、ペンを取り付けたヘッド部分を G-code で制御することで絵を描くというものであるが、プロッタをミシンの送り台の上に通し、ヘッド部分に枠を固定することで、データ通りに縫うことが可能になる。

4.2.1. ミシン針との連動

本節では、本ミシンがどのようにして動作するのかを説明したい。

本ミシンは、ボタンなどによる入力ではなく、フットコントローラーによる操作で扱うことができる。具体的には、フットコントローラーに圧力をかけることで、内部のスライドボリュームのつまみが移動して抵抗値が変化し、出力電圧がアナログに変化する。それによってフットコントローラーに接続されたミシン内部のモータに電力が伝わることで、針が上下に動作する仕組みになっている。このように、本ミシンはア

ナログ機械であることから、デジタル信号によって制御するためには専用のシステムの補完が必要となる。

その解決策としては、ミシン内部のモータを Arduino に接続して直接制御する方法と、電源を供給しているフットコントローラーの電力を制御する方法の二通りが考えられる。しかし、前者の場合はミシンの筐体をすべて解体しなければならない。これはたいへん手間を要するうえに、元に戻すことが困難になることから、フットコントローラー側に制御を設ける方法を選択するにいたった。

フットコントローラーの先端は AC プラグになっており、本来はコンセントに差し使用するものである。したがって、120V の交流電力を制御ボードの Arduino から制御しなければならないことになる。しかし、Arduino が高電圧の交流を直接あつかうことはできないことから、低電圧の信号によって制御できるソリッド・ステート・リレーを挟むことにした。本制作では SHARP 社製の S108T02 という型番のものを使用する。

まず、延長コードの中間にソリッド・ステート・リレーの高圧側端子を接続し、制御側の端子を Arduino に接続する。そして、その延長コードにフットコントローラーのプラグを接続し、Arduino の信号のオン／オフによってミシンへの電源供給を切り替えるようにした[Fig.27]。このように、Arduino からミシンの針の上下動作を行うことで、プロッタとの連動が可能になる。

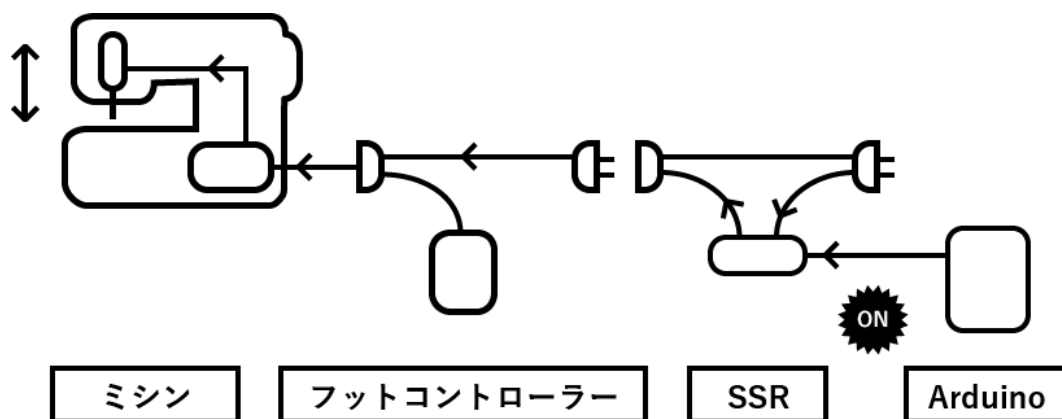


Fig.27 SSR 接続の図

電動ミシンの動力源であるモータは回転速度が不安定であるうえに、電動ミシン内部に制御システムが存在しないことから、針の停止位置は不定になってしまう。したがって、針が刺さった状態で布が動いた場合、布が裂けたり、針が折れたりする危険がある。このアクシデントを防ぐには、枠の移動中は針の停止位置を戻すという解決策が考えられる。

とはいえ、針目を正確にデジタル制御するためには、針の高さを離散的に定義しなけ

ればならない。そのためには針の高さを検知するための機能が必要となる。そこで考えられたのが、マイクロスイッチのように接触を検知する方法や、ロータリーエンコーダーによってモータの回転数を計測する方法であった。しかし、再現性を重視する本制作においては、取り付けが簡単かつ正確な検出が可能なフォトインタラプタによるセンシングを採用することにした。

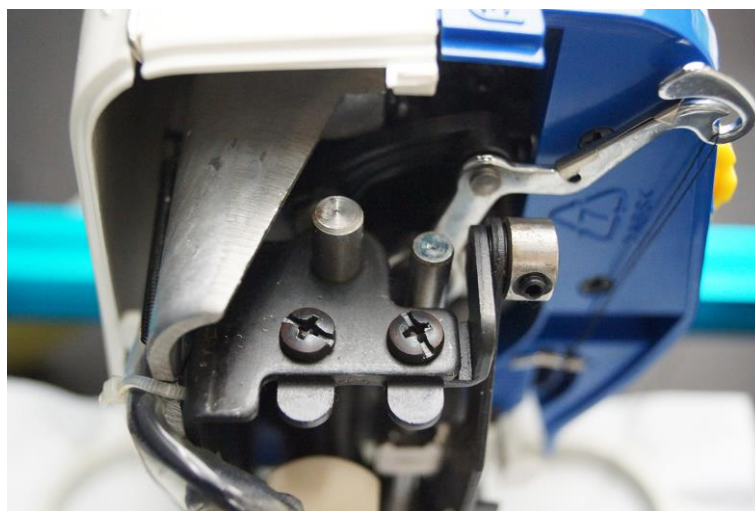


Fig.28 針棒とその周囲

フォトインタラプタは対向する発光部と受光部があり、両者の間を遮光することによって出力が変化するセンサである。本制作においては、Letex 社製の LBT-131 という型番を用いた。

針の位置の検出にあたっては、針棒の位置に着目する。針棒は針に上下の力を与える部位であり、針棒が上にあるときは針も上に、針棒が下にあるときは針も下にある。すなわち、針棒の位置を検出することは針の位置を検出することに等しい。また、ミシン内部の針棒上部には空間が設けられており、センサを取り付けても動作に影響はない。そして何より、針棒が存在する本体左上のカバーはネジひとつで取り外し可能であることから、着脱も容易である。以上の理由から、フォトインタラプタによる針棒の位置検出を行うことにした。

正確に位置を検出するために、針棒の上端にはフォトインタラプタを通過するかぎ形のパーツを取り付け、針棒が最上部にある場合に通過するように、フォトインタラプタ側にブラケットを取り付けて固定した[Fig.29]。これによって、安定して針の位置を検出することが可能となった。

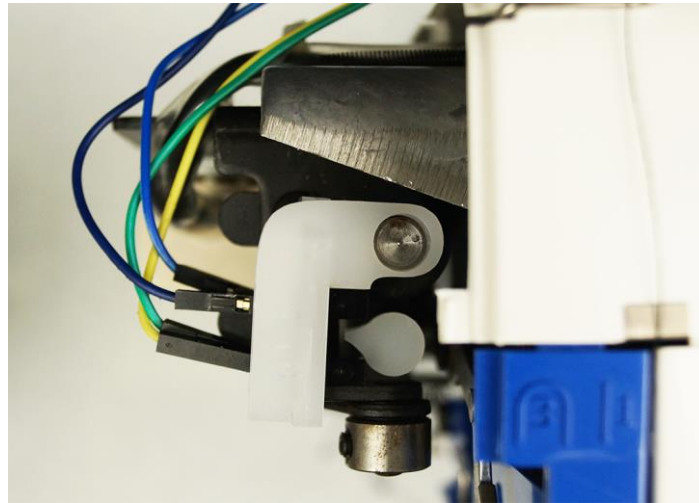


Fig.29 ブラケット

4.2.2. プログラム

前述の通り、本制作に使用しているプロッタのファームウェアは Reprap をベースにしたものである。Arduino はこのファームウェアを基本に、デジタルミシンに対応させるための変更を行う。

Reprap に採用されている制御言語の G-code は、もともとは NC マシンの制御のための NC プログラムを 3D プリンタ用に拡張したものである。G-code は、まず命令の種類を行頭に記述し、続いてパラメータを記述するという文法をとる[Fig.30]。例えば、「G00 X50.0 Y50.0 F3500.0」と入力されている場合、G00 は位置決めのための命令であるので、X 座標の 50.0、Y 座標の 50.0 まで移動するという意味となる。このように、G-code は人間でも直感的に理解できる制御言語となっている。

```
; G-code  
  
G21  
G00 X50.0 Y50.0 F3500.0  
G01 Z10.0 F5000.0 (Penetrate)  
G01 X100.0 Y50.0 Z10.0
```

Fig.30 G-code の記述例

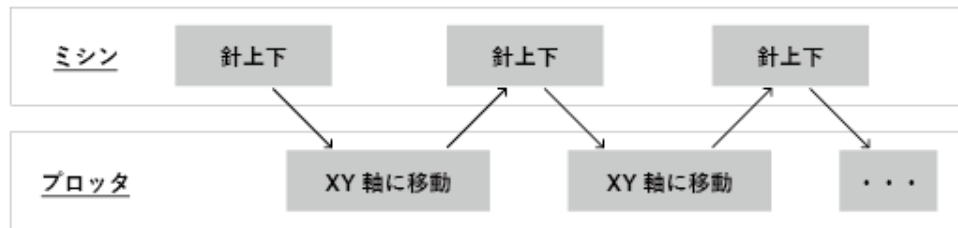


Fig.31 デジタル制御マシンの動作フロー

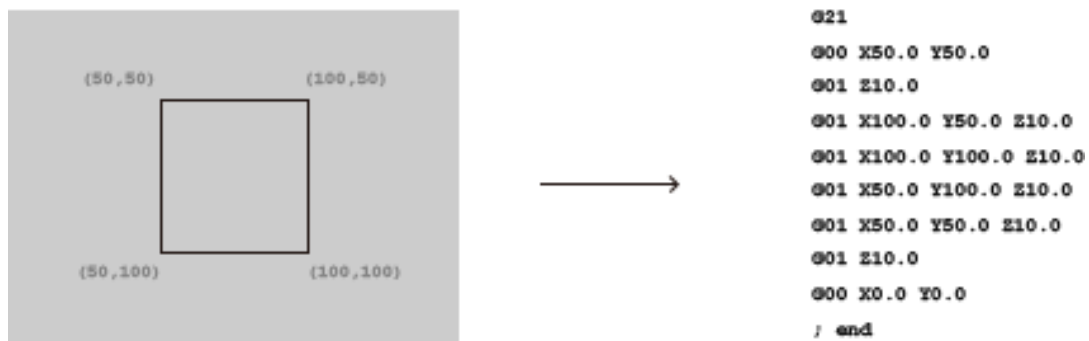
Fig.31 に、Fig.30 の命令を受けた針の動作の流れを図示した。このように、針の上下とプロッタの XY 軸移動を交互に行うことで、G-code による操作が果たされる。しかし、本来の G-code にはマシンを動作させるための命令は用意されていない。そこで必要になるのが、ファームウェアに変更を加えてマシンの動作を制御することである。今回の変更点は、G00 と G01 コマンドにおける振る舞いである。3D プリンタでは送り出しにあたる直線補完のコマンド G01 では、マシンの針を上下したあとにプロッタの XY 軸を移動させる。3D プリンタでは位置合わせにあたる早送りのコマンド G00 では、マシンの針は上下せず、XY 軸の移動だけを行う。詳細は付録のファームウェアのソースコードを参照いただきたい。

マシンの針の位置はフォトインタラプタによってセンシングしていることから、針棒が最上部から下へ降り、また最上部に戻ってくるまでを一針として、フィードバックによる制御を行っている [Fig.32]。



Fig.32 制御の流れ

このマシンのための G-code を生成するプログラムは Processing で実装した。前述の通り、マシンによる縫い目は線分の集合であると考えられる。シミュレーション画面上に描かれた直線を直線の始点、終点の座標に変換し、G-code の命令に割り当てる [Fig.33]。



シミュレーション画面

生成される G-code

Fig.33 Processing での G-code 生成

Fig.33 を例に、デジタル制御ミシンのための G-code の書式を説明する。一行目の G21 は、単位を mm に指定する命令である。そして次の行の G00 で位置合わせのために移動する。そして、三行目から縫い始めるわけであるが、最初に糸を確実に縫い留めるために同じ場所に二度針を差すようにしている。その後は G01 によって針の上下と移動を繰り返し、最終行で原点に戻ることで動作は終了する。

この一連の命令を G-code ファイルとして書き出し、ファームウェアの書き込まれた Arduino へ送信することでマシンが動作する。ソースコードは付録に記載している。Arduino への G-code 送信は、Processing のシリアル通信で可能である。また、必要に応じて Repetier-Host などの Reprap 向けホストソフトウェアを使用することもできる。



Fig.34 動作の様子

4.2.3. 布を固定する枠

本研究は、従来のコンピュータ刺繍ミシンを参考に、布を固定した枠を XY 方法に移動させる方法を採用している。当然ながら、布は張りを持たせた状態で固定しなければわんでしまい、正確に縫うことができなくなってしまう。従来のコンピュータ刺繍ミシンにおいては、手刺繍でも用いられる外枠と内枠の間に布をはめ込むタイプの刺繍枠を採用している。

しかし、張りを持たせた状態を維持することさえできれば、枠はどのようなものを用いてもよい。実際に手刺繍の分野においては、あらゆる刺繍枠の自作方法が模索され、ウェブ上でもその成果が公開されている。しかし、本研究においては従来のコンピュータ刺繍ミシンの刺繍枠を採用した。



Fig.35 刺繍枠

5. 実験・評価

5.1. デジタル制御ミシンの動作実験

本章においては、筆者の制作したデジタル制御ミシンを実際に動作させた実験の評価を行う。具体的には、G-code およびそのシミュレーション画面と、G-code によって実際に出力されたものを比較対照していく。

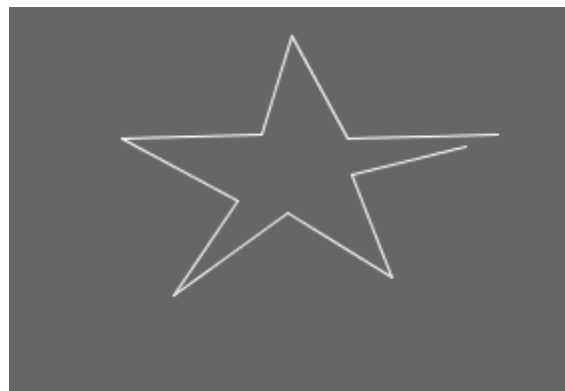


Fig.36 シミュレーション

```
G21
G00 X11.5 Y15.75
G01 Z10.0
G01 X30.25 Y16.25 Z10.0
G01 X37.25 Y3.5 Z10.0
G01 X41.0 Y15.75 Z10.0
G01 X58.5 Y16.25 Z10.0
G01 X44.0 Y24.0 Z10.0
G01 X52.0 Y35.75 Z10.0
G01 X37.75 Y25.5 Z10.0
G01 X24.75 Y33.5 Z10.0
G01 X29.75 Y20.75 Z10.0
G01 X15.5 Y17.25 Z10.0
G01 Z10.0
G00 X0.0 Y0.0
; end
```

Fig.37 Gcode

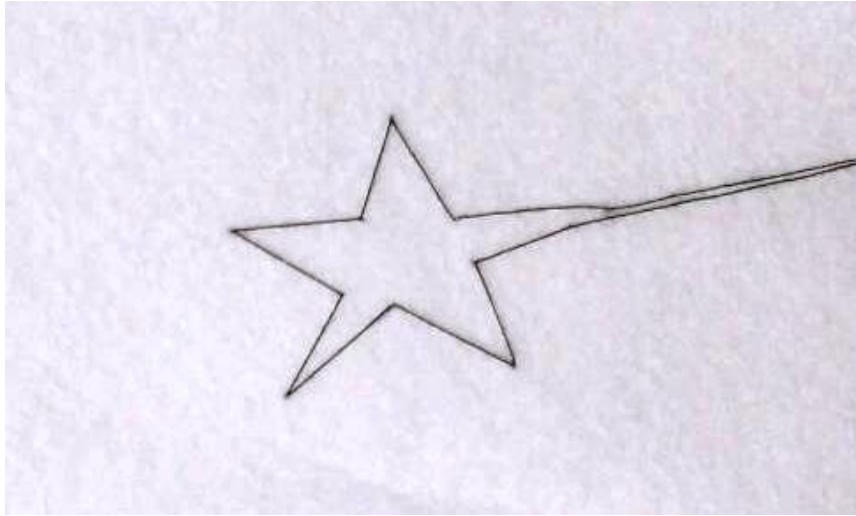


Fig.38 実際に作られたもの

上図は順番に、G-code 生成ソフトウェアで図形を描いたキャプチャと、そこから生成された G-code、そしてその G-code から実際につくられたものの写真に対応している。

本実験で使用した生地は、厚み 1mm のポリエステル製のフェルトである。実際に縫われた図形の大きさは約 40mm*20mm であり、動作開始から終了までの時間は 15 秒であった。

シミュレーションと実際につくられたものとを比較すると、おおむねデータの再現性は高いといえるだろう。針は一針ごとに停止し、針が下に刺さったまま動くといったエラーもなく、上下に一往復してからプロッタが XY 軸に移動するという想定通りの動作を観察することができた。シンプルなサンプルではあるが、この実験からミシンのデジタル制御化に成功したといえる。

5.2. 関連プロジェクトとの比較

ミシンをデジタル制御化するプロジェクト自体は、本研究のほかにも存在する。最後にそれらの比較対照を行うことで、実験と評価をしめくりたい。

まずは、LEGO マインドストームを用いてミシンを改造し、デジタル制御ミシンを制作しているプロジェクト「NXT Embroidery Machine」があげられる。これは csv ファイルから XY 軸の座標値を読み込んだうえで、既存のキットとミシンを用いて改造をしている点が本研究と類似している。枠に取り付けられた生地が XY 方向に移動する機構は同様であるが、はずみ車に取り付けたモータを回転させることで針の上下を制御すると機構は異なっている。

本研究と比較すると、LEGO マインドストームキットの制約から加工可能サイズが限られることと、モータのトルクが弱いことから針の運動に時間がかかるといった問題点があげられる。一方で、ミシン自体にはまったく改造を加えていないことから、本研究よりも容易に実践することができる。しかし、当プロジェクトはプログラムこそ公開されているが、LEGO の組み立て方が公開されていないという問題がある。

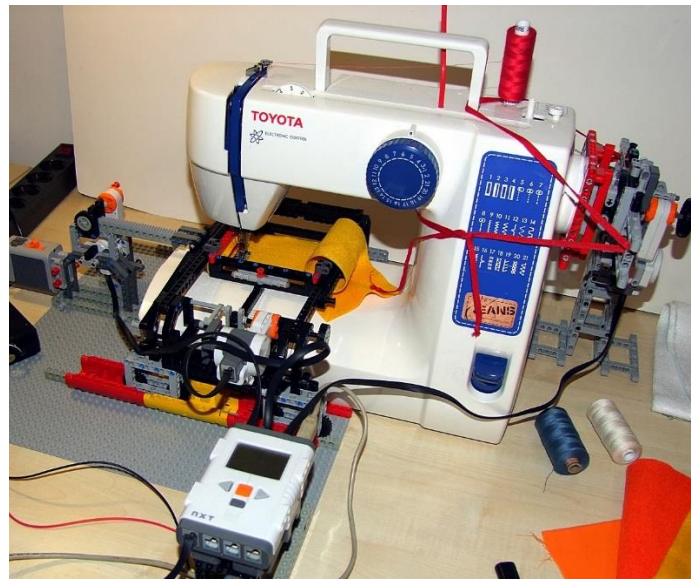


Fig.39 NXT Embroidery Machine

また、Disney Research Pittsburgh によるプロジェクト「Printing Teddy Bears: A Technique for 3D Printing of Soft Interactive Objects」は、立体的に毛糸を縫うための機械を Reprap ベースで開発している。従来のミシンのように上糸と下糸を絡ませるのではなく、毛糸を針で刺し込むことで絡ませて成形するニードルフェルトを自動化した機械である。したがって、ミシンとは違って毛糸の層を積層することで立体造形が可能であり、内部に異なる物質を挟んだ状態で縫い留めることもできる。

しかし残念なことに、NXT Embroidery Machine がこの機械のオープン化を視野に入れている一方で、論文と作り方を公開している点を除けば、Disney Research Pittsburgh にその意志は見られない。[12]

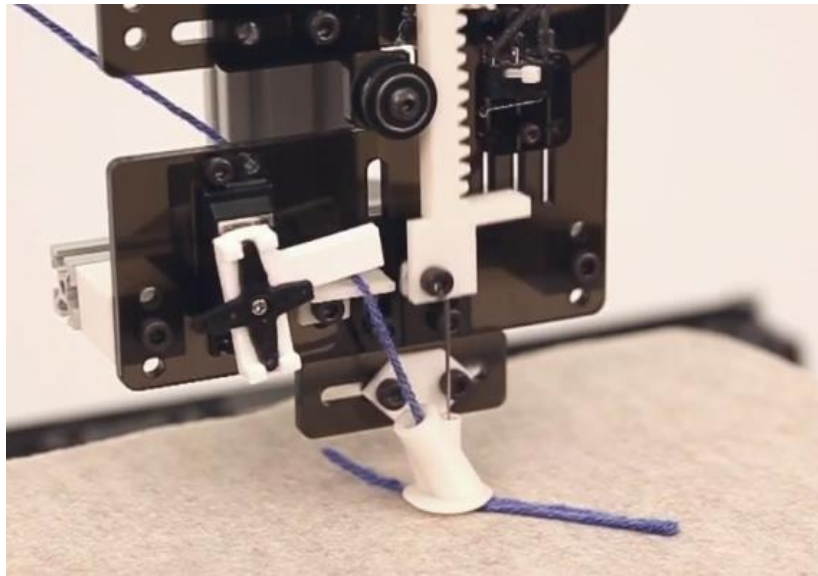


Fig.40 Printing Teddy Bears

6. 結論・展望

6.1. オープンな機械から生まれる新しいものの作り方

本研究では、既存のミシンを改変してデジタル制御化する手法を提案してきた。しかし、デジタルデータによって制御されるミシンとしては、コンピュータ刺繍ミシンという前例がすでに存在していた。そこで本節では、筆者が2章で提示したオープンなものづくりの条件を踏まえつつ、制作したデジタル制御ミシンを分析してみたい。

1. 設計が改変可能、かつ公開されている（機械の要件）

本研究におけるデジタル制御ミシンが従来のコンピュータ刺繍ミシンと異なるのは、自在に改変することができる機械であるという点が挙げられる。その変化は、オープンソースの現状さながらハードウェア／ソフトウェアの両面からとらえることができる。ハードウェア面においては、プロッタのサイズをスケールしてインテリアファブリックや建築に応用したり、ミシン糸より太い糸を扱うために針棒にアタッチメントを加えたりといった改変が考えられる。また、ソフトウェア面においては、3章で示した点字に最適化された G-code 生成ソフトウェアを実装することで、Illustrator のようなデザインソフトを介さずに操作することが可能になるだろう。自動縫製の例も、洋服の型紙データを変換するソフトウェアを実装すれば、簡単に衣服をつくることができるようになる。このように、本研究におけるデジタル制御ミシンと想定外の利用を良しとしないコンピュータ刺繍ミシンとを比較すると、非常に多様性に富んでいることがわかる。Reprap の歴史からも読み取れるように、機械の派生可能性こそ、つくられるモノの多様性を担保するファクターに他ならないのである。

2. 他者にシェアできる（データの要件）

従来のコンピュータ刺繍ミシンは、データの共有に問題を抱えていた。具体的には、メーカーごとに異なるデータ形式を採用しているために、メーカーを越境した刺繍データの共有ができないというデメリットを抱えていた。また、個人がみずからデータを作成すること自体、一般家庭で起こることが想定外であるという認識は、データ作成ソフトの法外な値段からも明らかである。これはいわば、コンピュータ刺繍ミシンの応用可能性をメーカーみずから潰している現状を裏付けるものだ。

しかし、本研究で採用した G-code であれば、制御データの共有も G-code 生成ソフトウェアの共有も容易である。また、ネット上のハウツーを参照すれば、初心者でも

簡単に、なおかつ無料で改変することができる。

以上のように、本研究におけるデジタル制御ミシンは、裁縫、手芸の分野におけるオープンなものづくりを実現させるための要件のうち、「機械の要件」と「データの要件」を満たしていることが確認できる。したがって、残りの「環境の要件」の実現について考えることが、本研究における目下の課題となるだろう。

6.2. 展望

3. 共有の可能な場がある（環境の要件）

従来のコンピュータ刺繍ミシンにおいては、その閉鎖的なデータ形式によってデータの共有が非現実的であった。しかし、その問題は本研究におけるデジタル制御ミシンにおいては解決されたと考えられる。

そこで次に必要となるのが、データを共有するためのプラットフォームである。オープンソースないしはオープンデザインの思想のもと、他者の参画と派生の連鎖を実現するには、連鎖を起こせるだけのユーザが集まる場を設ける必要がある。そのためには、ユーザーにとって魅力的なサービスやコミュニティを設計しなければならない。また、機械の作り方をどのように記述し、公開するのかという方針も重要な課題である。本研究が提示する未来像は、まずはデジタル制御ミシンが増殖しないことには始まることがない。

ことソフトウェアの場合はすべてがテキストで構成されていることから、ソースコードをコピーしてペーストすれば良い。しかし、実世界に物質として存在し、なおかる情報を扱わなければならないハードウェアは、3D データやプリント基板回路はデータ化されている一方で、機械の設計や配線といった手作業工程における統一されたデータ形式が存在しない。したがって、第三者に共有するためには、どのようなノートーションが適切か慎重に検討する必要がある。

オープンソース・ハードウェアと冠していながら、表記に対する配慮が欠けていることから失敗に終わってしまうプロジェクトも少なくない。オープンソース・ハードウェアとして成功するには、何よりもまず、他者による再現という前提に注意を払うことである。その記述にあたっては、テキストによる説明がないにも関わらず、低年齢の児童でも理解できるダイアグラムベースの設計がなされている LEGO の説明書なども参考になるだろう。

7. 謝辞

まずは研究会の指導者である田中浩也准教授に深くお礼を申し上げます。SFC に入り、デジタルファブリケーションに出会えたことは、私の人生に多大な影響を与えることとなりました。

田中浩也研究室のメンバーには、3 年間ものあいだ支えられてきました。まずは、研究室に入りたてで何もわからない私に多くのことを教えてくださった先輩方に感謝いたします。そして、ともに研究・制作をおこなってきた同期・後輩たちと切磋琢磨しながら過ごした時間もかけがえのないものでした。ありがとう。

水野大二郎准教授には、研究に関して多くの助言をいただきましたことをお礼申し上げます。同じ屋根のもとに近い研究分野で活動し、多くの刺激を与えてくださった水野研のメンバーにも感謝いたします。

そして何より、本研究のテーマは、慶應義塾湘南藤沢メディアセンターのファブスペースでデジタル刺繍ミシンに触れなければ、まったく違うものになっていたのではないかと思います。度重なるワークショップの開催を快諾してくださったことで、研究に与えられた影響は計り知れません。関係者のみなさまに感謝申し上げます。

最後に、4 年間の学生生活を温かく見守ってくれた家族に感謝して、謝辞といたします。

8. 参考文献

- [1] 田中浩也『SFを実現する——3D プリンタの想像力』講談社、2014
- [2] 伊藤穰一、スプツニ子！、ケヴィン・スラヴィン、ドミニク・チェン、藤井直敬、田中浩也、『角川インターネット講座(15) ネットで進化する人類 ビフォア／アフター・インターネット』角川学芸出版社、2015
- [3] RepRap.org , "RepRap - RepRapWiki", <http://reprap.org/> [アクセス日: 18 01 2015]
- [4] まつもとゆきひろ、法林浩之、久野靖、阿部和広、吉岡弘隆、やまねひでき、瀧田佐登子、鵜飼文敏『角川インターネット講座(2) ネットを支えるオープンソース ソフトウェアの進化』、角川学芸出版社、2015
- [5] クリエイティブ・コモンズ・ジャパン、"クリエイティブ・コモンズ・ライセンスとは | クリエイティブ・コモンズ・ジャパン"、<http://creativecommons.jp/licenses/> [アクセス日: 18 01 2015]
- [6] Bas van Abel, Roel Klaassen, Lucas Evers, Peter Troxler, Open Design Now, BIS publishers, 2011
- [7] アンドリュー・ゴードン『ミシンと日本の近代——消費者の創出』大島かおり訳、みすず書房、2013
- [8] 柏木博『日用品の文化誌』岩波新書、1999
- [9] Electroloom, "Electroloom", <http://www.electroloom.com/> [アクセス日: 18 01 2015]
- [10] FabTextiles, "Computational Couture @Beyond | FabTextiles", <https://fabtextiles.org/> [アクセス日: 18 01 2015]
- [11] Nervous System, "Kinematics Dress- Nervous System" <http://n-e-r-v-o-u-s.com/projects/sets/kinematics-dress/> [アクセス日: 18 01 2015]

[12] Scott E. Hudson, “Printing Teddy Bears: A Technique for 3D Printing of Soft Interactive Objects”, 2014, <https://www.disneyresearch.com/wp-content/uploads/Printing-Teddy-Bears-A-Technique-for-3D-Printing-of-Soft-Interactive-Objects-Paper.pdf> [アクセス日: 18 01 2015]

9. 図表引用元

Fig.1: Reprap

http://opensource.wikia.com/wiki/RepRap_Project

Fig.2: Reprap

http://reprap.org/wiki/RepRap_Family_Tree

Fig.3: Arduino

<https://www.arduino.cc/>

Fig.4: Repetier-Host

<http://www.repetier.com/>

Fig.5: Thingiverse

<http://www.thingiverse.com/thing:892654>

Fig.6: クリエイティブ・コモンズ・ライセンス

<http://creativecommons.jp/licenses/>

Fig.7: Wikipedia

<https://ja.wikipedia.org/wiki/%E3%83%9F%E3%82%B7%E3%83%B3>

Fig.8: Electroloom

<http://www.electroloom.com/>

Fig.9: Fabtextiles

<https://fabtextiles.org/>

Fig.10: nervous system

<http://n-e-r-v-o-u-s.com/projects/sets/kinematics-dress/>

Fig.21: コトバンク

<https://kotobank.jp/image/dictionary/nipponica/media/81306024002932.jpg>

Fig.24: IKEA

<http://www.ikea.com/jp/ja/>

Fig.25: Makeblock

<http://www.makeblock.cc/>

Fig.26: Makeblock

<http://www.makeblock.cc/>

Fig.39: LEGOISM

<http://www.legoism.info/2014/02/nxt-embroidery-machine.html>

Fig.40: Disney Research

<https://www.disneyresearch.com/project/printed-teddy-bears/>

付録

- ・ Reprap ファームウェア追加プログラム

```
//This code is for SewingMachineHack
//ssr conect to A0
int SSR_PIN = A0;

//photointerrupter connect to A7
int SENSOR_PIN = A7;

int sensorVal = 0;
int thresholdVal = 0;

void init_sewing()
{

    pinMode(SSR_PIN, OUTPUT);
    pinMode(SENSOR_PIN, INPUT);

    thresholdVal = 800;
    //code for setting threshold value will be here..

}

void ssr_control()
{

    do {
        sensorVal = checkSensor();
        digitalWrite(SSR_PIN, HIGH);
        delay(10);
    } while (sensorVal == 0);

    do {
        sensorVal = checkSensor();
```

```
        digitalWrite(SSR_PIN, HIGH);
        delay(10);
    } while (sensorVal == 1);

    digitalWrite(SSR_PIN, LOW);

}

int checkSensor() {
    if (analogRead(SENSOR_PIN) > thresholdVal) {
        return 1;
    } else {
        return 0;
    }
}
```

- ・ G-code 生成プログラム(Processing)

PrintWriter output;

```
int length = 100;  
float[] posX = new float[length];  
float[] posY = new float[length];
```

```
float cposX;  
float cposY;
```

```
float a;  
float b;
```

```
int n = 1;  
int first = 0;
```

```
String startcode = "G21";  
String textValue = "test";
```

```
void setup() {  
    PFont font = createFont("arial",20);  
  
    textFont(font);  
  
    frameRate(30);  
    size(600, 400);  
  
}
```

```
void draw() {  
    stroke(255);  
    background(102);  
    fill(255);
```



```

text(cp5.get(Textfield.class,"input").getText(), 360,130);
text(textValue, 360,180);

if (mousePressed) {

    cposX = mouseX;
    cposY = mouseY;

    if ((cpoSX != posX[(n-1)]) || (cpoS != posY[(n-1)])) {

        a = abs(cposX - posX[n-1]);

        b = abs(cposY - posY[n-1]);

        if (first == 0){
            posX[0] = cposX;
            posY[0] = cposY;
            first ++;
        } else if (a < 10 && b < 10){
            println(a + " " + b);
        } else {

            posX[n] = cposX;
            posY[n] = cposY;

            println("x:" + posX[n] + " y:" + posY[n]);

            n = n + 1;
        }

    }

}

for (int i=0 ; i < n-1 ; i++){
    line(posX[i], posY[i], posX[i+1], posY[i+1]);
}

```

```

    }

    if ((keyPressed == true) && (key == 's')) {
        exportFile();
    }
    if ((keyPressed == true) && (key == 'd')) {
        for (int j=0; j < n; j++){
            posX[j] = 0;
            posY[j] = 0;
        }
        n = 1;
        first = 0;
    }
}

// export g-code file

void exportFile() {

    output = createWriter("gcode" + month() + day() + hour() + minute()
+ ".gcode");

    //Insert startcode
    output.println(startcode);

    output.println("G00 X" + posX[0] + " Y" + posY[0]);
    output.println("G01 Z10.0");

    for (int k=1; k < n; k++){

        output.println("G01 X" + posX[k] + " Y" + posY[k] + " Z10.0");

    }

    output.println("G01 Z10.0");

```

```
        output.println("; end");  
        output.flush();  
        output.close();  
    }
```