

Title	ロボット設計のためのスケッチベースインターフェース
Author(s)	品田, 康平
Citation	
Issue Date	2026-03
Type	Thesis or Dissertation
Text version	author
URL	https://hdl.handle.net/10119/20492
Rights	
Description	Supervisor:謝 浩然, 先端科学技術研究科, 修士(知識科学)

修士論文

ロボット設計のためのスケッチベースインターフェース

品田 康平

主指導教員 謝 浩然

北陸先端科学技術大学院大学
先端科学技術専攻
(知識科学)

令和8年3月

Abstract

In recent years, robotics research has actively focused on developing robots that assist human daily life tasks, such as folding clothes and brewing coffee. While these efforts aim to improve people’s lives by introducing robots into households, current robots remain expensive and difficult to adopt. One potential solution is personal robot development. The widespread availability of consumer-grade 3D printers has made it easy to fabricate 3D models at home or in schools. In the robotics field, 3D-printed robots are commonly used for evaluation experiments, as they can be manufactured at low cost. However, personal robot development has not yet become widespread. This is because robot design requires extensive specialized knowledge spanning mechanical engineering, electrical engineering, and computer science. Furthermore, there are issues with existing design methodologies. Current approaches are polarized between CAD-based and modular methods. CAD-based methods use 3D modeling software and, while complex and difficult to operate, enable free and precise robot design through numerical calculations and inputs. Modular methods involve arranging and connecting predefined parts, offering simple operation but limited design freedom due to constrained design spaces. Consequently, it is difficult for non-expert users to design robots that match their intentions. Sketch-based interfaces offer an effective solution to this polarization problem. Sketching is a time-honored form of expression familiar to many people, enabling intuitive operation. In the HCI field, sketch-based interfaces have been employed in design support systems, where user sketches are processed by algorithms or neural networks to handle specialized tasks such as calculations and analysis, thereby supporting design by non-experts. Robot design encompasses the challenging issues for both shape and structure. A 3D model alone cannot function as a robot; joint information is essential for operation. Conventional methods require either precise numerical definitions or rely on modules with predefined joint roles. To support robot design by non-expert users, methods that allow intuitive and flexible joint definition are necessary.

In this paper, I address the problem that robot design methodologies are polarized, and no system exists that combines intuitive operation with design freedom for both shape and structure design by non-expert users. I developed sketch-based interfaces for robot design. First, I describe Sketch2Joints, a system that supports joint structure definition for non-expert users through joint symbolization. In Sketch2Joints, users place symbolized joints on a canvas and connect them with straight or curved lines to input joint structure information. Based on this input, the system defines a kinematic model and provides kinematic feedback by displaying links as line segments and updating the posture display according to joint angles. The system also automatically generates links for the designed joint

structure, which can be 3D-printed. By connecting these with pre-designed joints, users can fabricate their designed robots. Next, I describe Sketch2Robpla, a system that enhances shape freedom and kinematic feedback from Sketch2Joints while replacing joint information input with more intuitive operations. Sketch2Robpla takes user sketches as input and generates 3D robot models using generative AI. This enables users to design free-form shapes without learning 3D CAD or requiring complex numerical inputs. Joint information is then input by segmenting the generated shape with line strokes. Users draw lines over the 3D model view, and the system calculates cutting planes using the lines and camera viewpoint to segment the model, enabling intuitive model division. The system automatically places joints at the center of each cross-section and computes joint positions, rotation axes, and connection relationships. This automation reduces user burden, and kinematic feedback reflecting the actual shape allows users to visually understand the completed design. Additionally, physical joints can be inserted into the robot model. While existing methods require designing mounting positions and wiring plans according to motors, Sketch2Robpla reduces user burden by placing pre-designed meshes corresponding to motors at joint positions aligned with rotation axes, removing interfering portions, and converting the model into a robot model with joint mechanisms. Finally, the system outputs a URDF(Unified Robot Description Format) file describing the robot’s shape and structure. The URDF file consists of text representing the structure and meshes representing each link’s shape. The output meshes can be loaded into 3D slicer software, converted to G-code, and 3D-printed. User experiments confirmed that Sketch2Robpla supports robot design by non-expert users and achieves both intuitive operation and design freedom.

目次

第1章	はじめに	1
1.1	研究背景	2
1.1.1	人々の暮らしとロボット	2
1.1.2	既存のロボット設計手法	2
1.1.3	スケッチインターフェース	3
1.2	研究目的	4
1.3	本論文の構成	5
第2章	関連研究	6
2.1	スケッチインターフェース	6
2.2	3D モデル生成	10
2.3	既存のロボット設計手法	14
第3章	提案フレームワーク	19
3.1	全体フレームワーク	19
3.2	運動学シミュレーション	19
3.3	URDF(Unified Robot Description Format)	21
3.4	条件付き画像生成	21
3.5	3次元モデル生成	22
第4章	Sketch2Joints	23
4.1	手法	23
4.1.1	設計インターフェース	23
4.1.2	ユーザフィードバック	25
4.1.3	関節の製作	25
4.2	結果	29
4.2.1	製作	29
4.2.2	ユーザスタディ	30
4.2.3	動作実験	31
第5章	Sketch2Robpla	39
5.1	手法	39
5.1.1	形状生成インターフェース	39

5.1.2	関節付加インターフェース	44
5.2	結果	52
5.2.1	ユーザ実験	52
5.2.2	製作	53
第 6 章	おわりに	62
6.1	課題	62
6.2	展望	62
第 7 章	謝辞	64

目 次

1.1	研究概要図	4
2.1	イラスト制作支援システム AniFacedDawing[6]	6
2.2	3D 点群データを入力としたスケッチベースの 3D 形状生成 [7]	7
2.3	スケッチから流体シミュレーションシステム DualSmoke[8]	7
2.4	スケッチベースリンク機構作成システム Computational design of mechanical characters[10]	8
2.5	スケッチベースアニメーション付き 3D メッシュ作成システム Monster Mash[11]	8
2.6	スケッチベースタスク指示システム SketchMoMa[12]	9
2.7	単視点画像入力による 3D 生成システム TRELLIS[13]	10
2.8	パーツ別の 3D 生成システム OmniPart[15]	10
2.9	3D メッシュをテキストトークンとして扱う 3D 生成システム LlaMa-Mesh[16]	11
2.10	プリミティブメッシュを用いた 3D 生成システム MeshLLM[17]	11
2.11	3D 点群データを LLM で学習した 3D 生成システム 3D-LLM[18]	12
2.12	多視点画像を学習した 3D 生成システム ShapeLLM[19]	13
2.13	3D ボクセルを学習した 3D 生成システム ShapeLLM-Omni[20]	13
2.14	折り紙モジュール型ロボット設計システム Interactive Robogami[4]	14
2.15	3D プリント可能なロボット設計システム Interactive design of 3d-printable robotic creatures[21]	15
2.16	グラフ文法ロボット生成システム RoboGrammar[22]	16
2.17	進化的アルゴリズムによるロボット最適化システム RoboMorph[23]	16
2.18	ロボット自動生成システム RoboGen[24]	17
2.19	LLM によって報酬関数を決定するロボット設計システム Eureka[26]	17
2.20	LLM がタスク分割と報酬関数を決定するロボット設計システム Text2Reward[27]	18
3.1	提案システム Sketch2Joints と Sketch2Robpla の比較	20
3.2	URDF の例	21
4.1	提案システム Sketch2Joints 概要図	23
4.2	提案システム Sketch2Joints のインターフェース	24
4.3	関節編集フレーム	25

4.4	シミュレーションインターフェース	26
4.5	Motor タブ	27
4.6	Axis タブ	27
4.7	関節	27
4.8	図 4.2 のスケッチから作成したリンク	28
4.10	Sketch2Joints の入出力実験結果 1	32
4.11	Sketch2Joints の入出力実験結果 2	33
4.12	製作したロボットアームの各要素	34
4.13	被験者の入力した関節構造	35
4.14	内部構造 (左) と内部構造挿入後のぬいぐるみ (右)	36
4.15	System Usability Scale に基づくアンケートの結果	37
4.16	図 4.2 のスケッチから作成したロボットアーム	37
4.17	物体を運ぶ実験の様子	38
4.18	ロボットアーム自身の移動実験の様子	38
5.1	提案システム Sketch2Robpla	40
5.2	形状生成インターフェース概要図	40
5.3	形状生成インターフェース	41
5.4	生成設定	42
5.5	生成例 1	43
5.6	生成例 2	43
5.7	関節付加インターフェース概要図	44
5.8	関節付加インターフェース	45
5.9	関節情報フレーム	46
5.10	詳細編集ダイアログ	46
5.11	関節メッシュプルダウンリスト	47
5.12	full_Horizonal_Joint.stl	48
5.13	full_vertical_joint.stl	48
5.14	horizontal_miyajima_parent.stl	48
5.15	horizontal_miyajima_child.stl	48
5.16	vertical_miyajima_parent.stl	48
5.17	vertical_miyajima_child.stl	48
5.18	干渉除去	49
5.19	運動学フィードバック	51
5.20	ユーザ A による製作結果	54
5.21	ユーザ B による製作結果	55
5.22	ユーザ C による製作結果	56
5.23	ユーザ D による製作結果	57
5.24	ユーザ E による製作結果	58

5.25 ユーザ F による製作結果	59
5.26 System Usability Scale の結果	60
5.27 NASA-TLX の結果 (平均)	60
5.28 NASA-TLX の結果 (項目別)	61
5.29 製作結果	61

第1章 はじめに

本論文では、スケッチインターフェースによるロボット設計システムを提案する。提案システムは、専門的知識を持たないユーザのロボット設計を目的とし、従来の複雑な数値入力を、老若男女問わず多くの人間が親しみやすいスケッチ操作に置き換えることで、直感的な操作でロボットを設計することができる。ロボットの形状の設計で主に使われる 3DCAD は習得に時間がかかり、初学者には難しいソフトウェアであるため、本論文ではユーザが描いたロボットのスケッチを入力とすることで、幅広いユーザの利用が可能である。ロボットの構造スケッチを入力とすることで運動学構造の設計が可能なインターフェースを開発し、開発したインターフェースを用いて、利用者がぬいぐるみ内部に挿入する運動学的構造を設計することでスケッチインターフェースが専門的知識を持たないユーザのロボット設計を支援することが可能かを確かめた。その後、ユーザのスケッチからロボットの 3D モデルを生成し、生成した 3D モデルにユーザが関節構造を付加することでロボットを設計できるインターフェースを開発した。被験者のスケッチから作成されたロボットを 3D プリンタを用いて実際に製作することで評価を行う。

本システムでは、ユーザは提案インターフェースを利用することで任意の形状、構造のロボットを設計することができる。スケッチベースの手法は老若男女問わず経験のある直感的な操作であり、これを用いることで専門的知識を持たないユーザもロボットの形状を設計できる、関節構造の入力も従来のパラメータ入力ではなくモデルに線分を引く操作によって行うことで容易に構造を設計することができる。設計中のロボット構造の確認は運動学計算が必要になるが、システムが計算して表示するためユーザはスライダで関節角度を操作することでロボットの構造とその動作を視覚的に理解できる。製作においてはモデルの関節部分を予め設計したモータの取付が可能な部品に置き換え、3D プリンタを用いてモデルをリンクごとに製作し、ユーザが組み立てる。これにより、ユーザの製作を支援する。

本章では、研究背景、研究目的、本論文の構成について記述する。

1.1 研究背景

1.1.1 人々の暮らしとロボット

人々の生活の中にロボットが存在する世界は、ドラえもんや鉄腕アトムの世界のような魅力的な世界であると同時に、少子高齢化や労働力不足という問題を抱える日本にとって目標とするべき世界である。ロボティクス分野で現在盛んに研究されている VLA(Vision Language Action) モデルは、人々の生活の中にロボットが存在する世界の構築という目標に大きく関係している。Physical Intelligence 社の開発した VLA モデル π_0 [1] は自らの有意性を示す実験として、衣類をたたんで見せた。それだけではなく、 π_0 は未見のロボットや環境においても、自然言語による命令でタスクが実行可能であることを証明した。これはロボットのオペレーターに専門知識を必要とせず、一般の人々でもロボットの制御を行える、すなわちロボット制御が民主化される未来の可能性を示唆している。

VLA モデルによってロボット制御の民主化がなされたが、一般家庭にロボットが普及するためにはロボットが高価だという問題がある。この問題はロボットを自作することで解決することができる。家庭用 3D プリンタの普及により、自宅や学校でも 3D モデルの立体化が容易になった。実際に評価実験を 3D プリンタ製のロボットで行う研究も少なくない。しかし、ロボット設計は機械工学、電気工学、運動学などの様々な専門的知識を必要とするため、複雑で参入障壁が高い。ロボットを一般家庭に普及させるためには、ロボット制御の民主化だけではなく、このロボット設計を民主化することが必要である。

また、ロボットの非専門家がロボット設計を行えるとすると、伝統産業の支援を目的としたロボットのような特殊な課題に対応するための特注のロボットを設計することができる。通常、ロボット設計者が専門家にヒアリングを行って予備知識を身に付けることが必要となるが、ロボット設計が民主化されていれば、伝統産業の従事者本人が自分のニーズを満たすロボットを自分で設計することが可能になり、伝統産業の存続に貢献できるだろう。

1.1.2 既存のロボット設計手法

既存のロボット設計手法は、CAD 型とモジュール型の二つに分けられる。

CAD 型: Autodesk Fusion[2] や SolidWorks[3] などの 3D モデリングソフトを用いてロボットの形状を設計する手法である。厳密な数値設定によって自由な形状を設計できる反面、ソフトウェアの操作習得に時間がかかる点が新規参入者を遠ざけている。また、ロボット設計はモデルの形状だけではなく、搭載されるアクチュエータ、センサ、それらを動作させるためのバッテリーや電気回路などのロボット構造に関わる要素についても設計する。機械工学、電気工学、制御工学について

の知識が必要となるため、非専門家が3Dモデリングソフトを用いてロボット設計を行うことは難しい。

モジュール型: Interactive Robogami[4] や V-rep[5] などをはじめとする予め作成した部品のデータベースから、部品を選んでロボットを設計する手法である。ロボット構造に関わる要素は容易されている部品に組み込まれていることが多く、CAD型と異なり非専門家でもロボットを設計しやすいが、限られた設計空間で設計を行うため、形状の自由度はCAD型に比べて低い。モジュールの数を増やして設計空間を大きくすることでこの問題は解決できるが、広大な設計空間はかえってユーザを困惑させてしまう。

現状のロボット設計手法の課題は、この手法の二分化である。ロボット設計の初心者にとって、モジュール型手法はロボット構造に関わる要素の設計がシステム化されており扱いやすいが、高い形状自由度を求める場合、CAD型の複雑な操作と他分野にわたる広範な知識が必要になる。高い形状自由度とロボットのための専門的な調整の自動化を両立することでロボット設計は民主化され、ロボット設計者の人口を増やし、ロボティクス分野の発展につながる。

1.1.3 スケッチインターフェース

スケッチインターフェースを用いたデザイン支援システムはHCI分野で多く研究されてきた。ユーザのラフスケッチからアニメ調の画像を生成してユーザの支援する AniFaceDrawing[6], 3D点群データとスケッチから3Dモデルを構成する研究[7], 流れや密度のスケッチから煙をはじめとするボリュームデータを作成する[8]。これらの研究の特徴は、従来必要だった専用ソフトの複雑な操作をスケッチインターフェースにより単純化し、解析、計算などの専門的な知識が必要な処理をシステムで自動化することでユーザのデザインを支援することである。近年では、ユーザの簡単なスケッチ入力を解析してGANをはじめとする生成AIによってユーザの入力を補完する形のシステムも多い。この手法をロボット設計システムに取り入れ、3Dモデリングソフトの複雑な操作をスケッチインターフェースによって単純化することで、システムを利用可能なユーザの範囲を拡げることができ、ロボット設計の民主化につながる。仮に、スケッチインターフェースを用いてロボット形状の設計が誰にでも可能になったとする。しかしこれによってロボット設計が誰にでもできるようになったかと言うと、そうではない。形状の設計だけでは、ロボットは動作することができない。CAD型では構造を形状に組み込んで設計することで構造を設計する。モジュール型ロボット設計システムにおいては、パーツが脚、胴などの役割を持っているため、それに基づいて構造を決定することができる。ロボット設計は、外観や重量などに関する形状の情報と関節による動きに関わる構造の情報の両方が必要になる。

1.2 研究目的

本研究では、既存のロボット設計手法の持つ形状自由度の高さと操作の容易さを両立する設計手法が存在しないという課題を解決するため、スケッチベースのロボット設計システムを提案する。はじめに、記号化したロボットの関節をユーザがキャンバス上に配置することでロボット構造を入力し、リンクを自動で作成して関節構造を設計するシステム Sketch2Joints について説明する。そして、Sketch2Joints で発見した運動学フィードバックの不足や形状自由度の低さ、関節構造入力の不明瞭さ等の問題点について改善を行ったシステム Sketch2Robpla を提案する。Sketch2Robpla の概要を図 1.1 に示す。Sketch2Robpla は、ユーザの入力したスケッチからロボットの形状を設計し、3D モデルにロボット構造を定義できるインターフェースによってユーザが関節構造を入力することでロボット設計を行う。低い操作難易度と高い形状自由度を両立するためにユーザのスケッチを入力とし、3D モデル生成 AI「TRELLIS[9]」を用いて 3D モデル化する。生成した 3D モデルをロボットとして扱うために、GUI を用いてユーザが関節構造を定義してロボット用のマークアップ言語である URDF 形式のファイルを作成する。そして URDF ファイルの関節部分を予め設計した関節メッシュに置き換え、モータの取付を可能にしたモデルを出力する。このモデルを 3D プリンタで印刷することでロボットを製作ができる。本研究では、提案システムによってロボット設計を民主化し、ロボット設計者の人口を増やし、これからロボットを扱うすべての分野に貢献することを目指す。

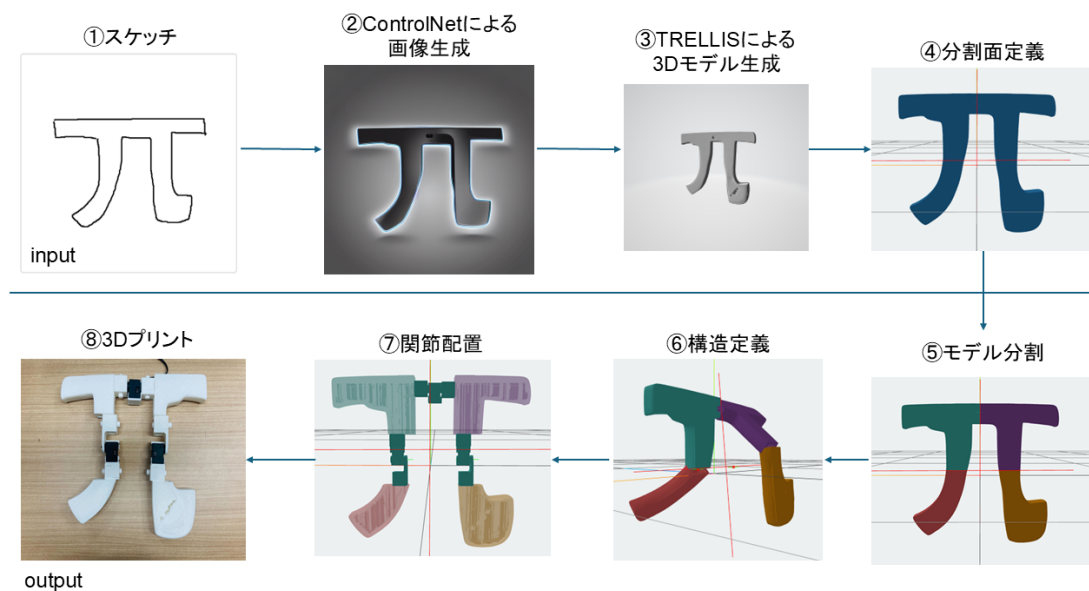


図 1.1: 研究概要図

1.3 本論文の構成

本論文ではまず，第1章にて研究背景，研究目的について述べた．第2章では関連研究について述べる．第3章ではこの研究の全体構造と予備知識について述べる．第4章でロボット構造の記号化によって関節構造の設計を実現した Sketch2Joints の手法と結果，評価について述べる．第5章では Sketch2Joints の改良として，3D 生成 AI による自由な形状生成とスケッチベースの操作で関節構造の定義を行うシステム，Sketch2Robpla の手法，結果，評価について述べる．第6章で，本研究のまとめと今後の展望について述べる．

第2章 関連研究

本研究は、スケッチインターフェース、3D モデル生成 AI、形態、行動最適化によるロボット設計の分野と関係しているため、これらについての関連研究を紹介する。

2.1 スケッチインターフェース

スケッチは古くから人間に親しまれてきた動作であり、スケッチインターフェースは老若男女問わず直感的に操作できることが利点である。特に HCI 分野でデザイン支援システムとして用いられることが多い。例えば AniFaceDrawing[6] は初心者のユーザのラフスケッチから StyleGAN を用いて高品質なアニメ調の画像を生成し、ユーザを支援する。書きかけの数本の線から高品質な画像を生成し、形状を変えずにスタイルや色を変更することもできるため、ユーザがどこに線を引くか、影を落とすかなど次の描画の参考になる。

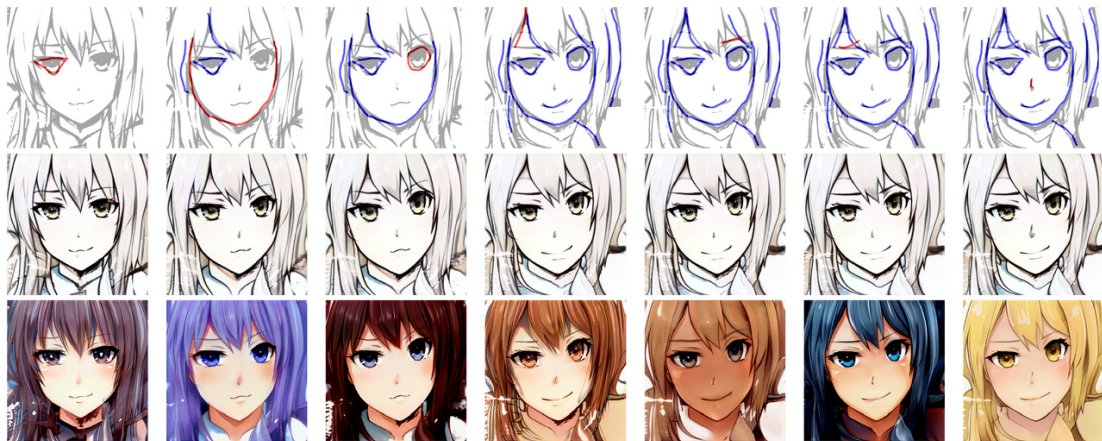


図 2.1: イラスト制作支援システム AniFacedDawing[6]

また、入力した点群データにユーザが輪郭をスケッチし 3D メッシュを生成するシステムも研究されている [7] スケッチからの生成では奥行の情報が入りず、点群からの生成では輪郭の情報が入りないという二つの手法の欠点を互いに補うため、スケッチと点群のどこが同期しているかを推論し、スケッチに一致するように点群を調整してメッシュを作成する。

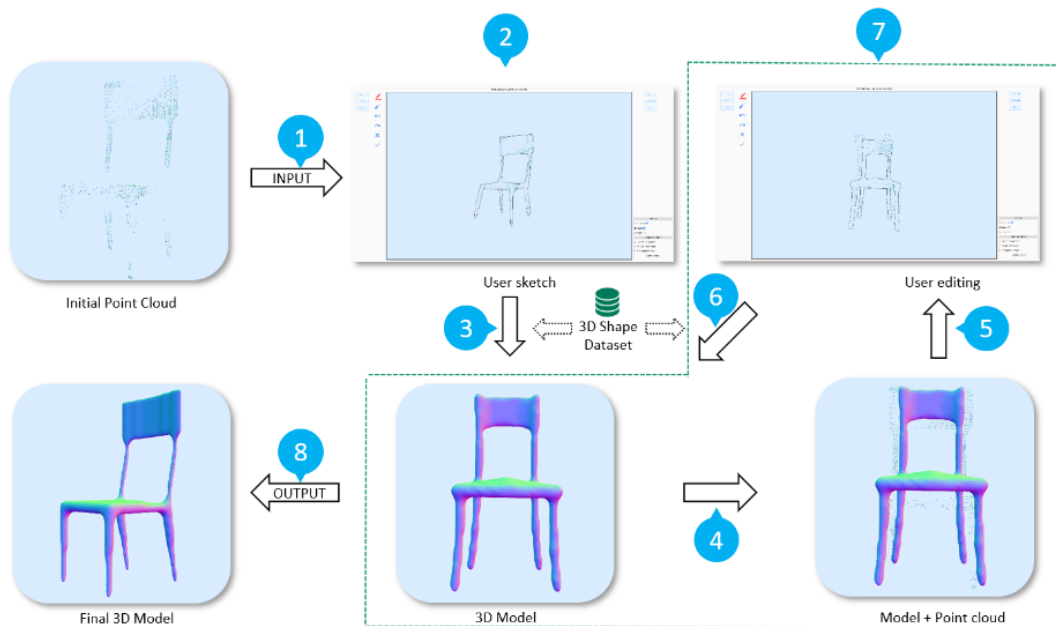


図 2.2: 3D 点群データを入力としたスケッチベースの 3D 形状生成 [7]

他にも、従来、時間のかかっていた煙のようなボリュームデータのレンダリングを、詳細な部分と大まかな部分の 2 種類の分類と DNN(Deep Neural Network)により高速化し、スケッチによる流れや密度の入力から、ボリュームデータのレンダリングを行うシステムである DualSmoke[8],

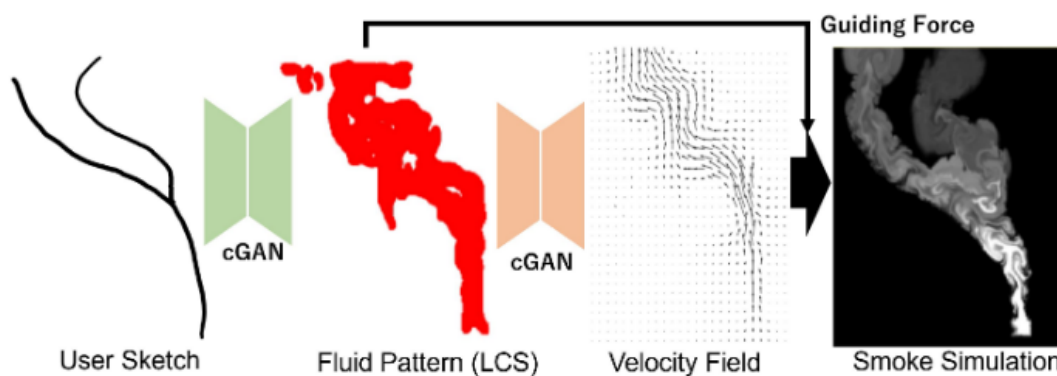


図 2.3: スケッチから流体シミュレーションシステム DualSmoke[8]

ユーザが画面上のキャラクターのパーツの動きをスケッチし、リンク機構のシミュレーション結果を集めたデータベースから、スケッチされた動きと最も近いリンク機構を検索、反復計算により調整してリンク機構のパラメータ (固定点, リンクの長さ等) を決定する. そして、確率的アルゴリズムを用いる焼きなまし法により、コンパクトで衝突しない歯車の組み合わせと配置を探索することで、スケッ

チからリンク機構を設計する研究 [10],

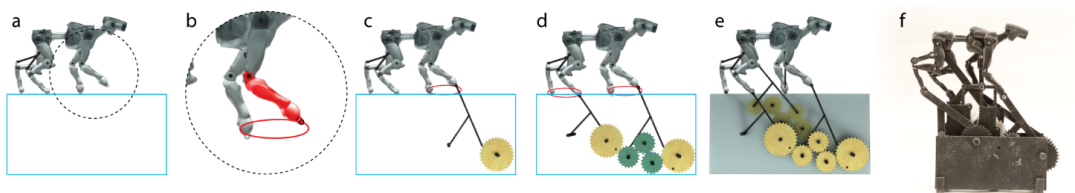


図 2.4: スケッチベースリンク機構作成システム Computational design of mechanical characters[10]

ユーザのスケッチを風船のように膨らませて 3D モデルを作成し、モデルの一点を掴んで動かすことで 3D モデルに変形アルゴリズム A-RAP(as-rigid-as-possible) 動きを与え、アニメーションを作成することができる MonsterMash[11] などがある。

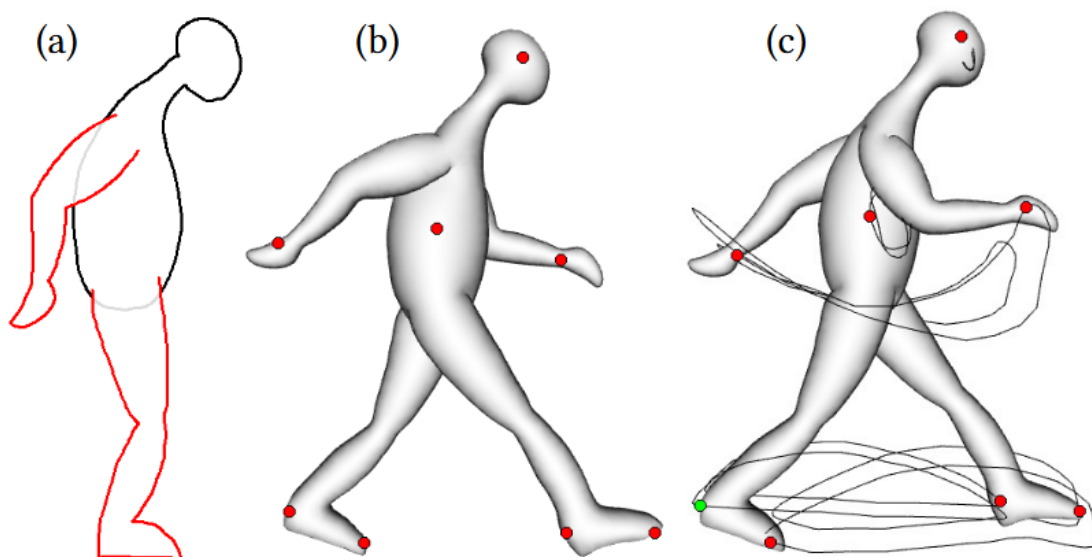


図 2.5: スケッチベースアニメーション付き 3D メッシュ作成システム Monster Mash[11]

さらにロボット分野におけるスケッチインターフェースとしてはSketch-MoMa[12]がある。Sketch-MoMa はロボットが取得した画像 (ロボットの頭や手首などに搭載されているカメラ画像) に対して、ユーザが運びたいものを丸で囲んで、移動方向に矢印を描く、ロボットの足下に移動経路を矢印で描くなどの命令をスケッチで行う。そして描かれたスケッチを VLM(Vision-Language Model) でタスクとして解釈し、それに基づいてロボットの動作を計画する。

これらの関連研究が示す通り、スケッチインターフェースは解析、計算などの専門的知識の必要な処理の自動化と組み合わせることで、非専門家のデザインを

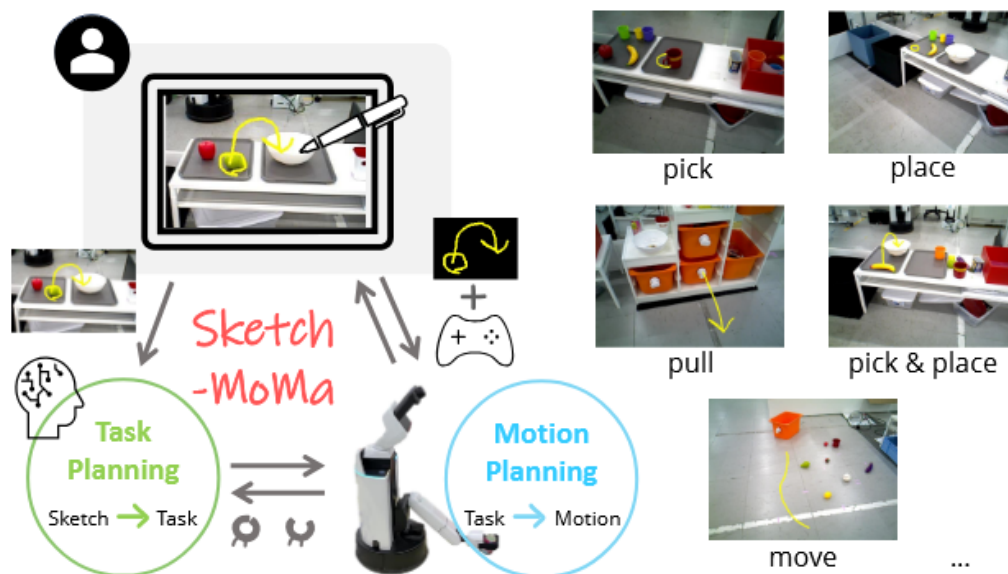


図 2.6: スケッチベースタスク指示システム SketchMoMa[12]

支援することができる．近年では，DNN や VLM と組み合わせることにより複雑なもののデザインをスケッチインターフェースで行えるようになった．

2.2 3D モデル生成

言語や画像の入力を基に 3D モデルを生成する技術は、拡散モデルと共に発展してきた。TRELLIS[13], Tripo[14] は単視点画像からの 3D モデル生成を行う研究である。また、OmniPart[15] は単視点画像からの 3D モデル生成に加えて、自動的な分割も行っている。

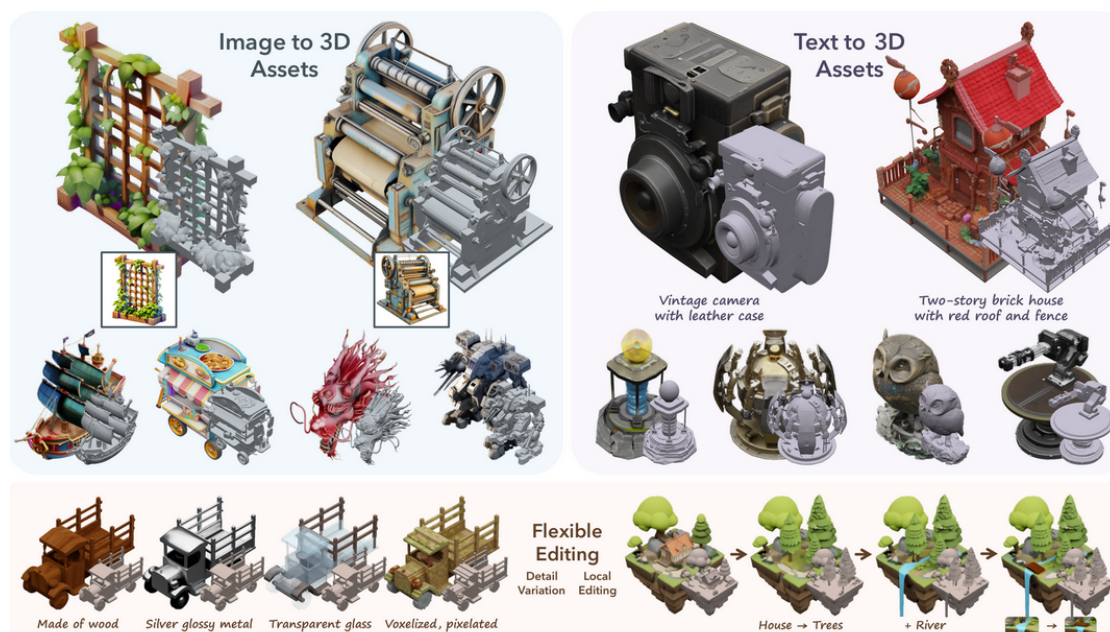


図 2.7: 単視点画像入力による 3D 生成システム TRELLIS[13]

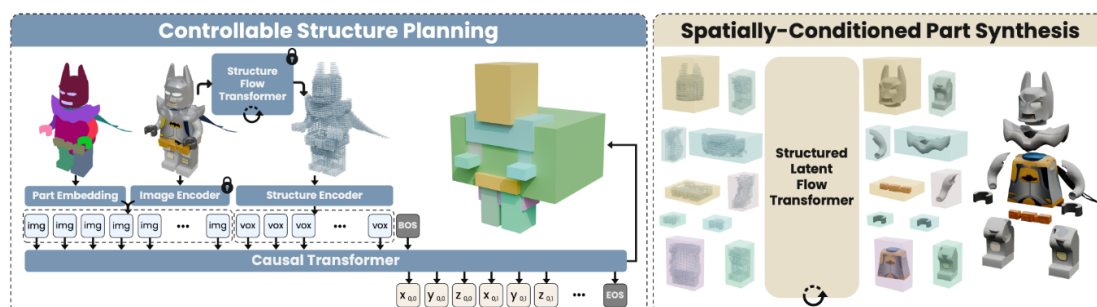


図 2.8: パーツ別の 3D 生成システム OmniPart[15]

本研究では、Trellis を用いて 3D モデルを生成することで、ユーザが 3D モデリングソフトを使わずに形状設計をできるようにした。また、近年では 3D モデルの形状を LLM で理解することで、3D モデル生成の品質を向上させる試みも多く見られる。LLaMa-Mesh[16] は、3D メッシュのプレーンテキスト表現をトークン化して LLM で学習することでテキスト生成の性能を維持しながら 3D メッシュ生成を実現し、3D メッシュの理解や解釈、対話的な生成を可能にした。

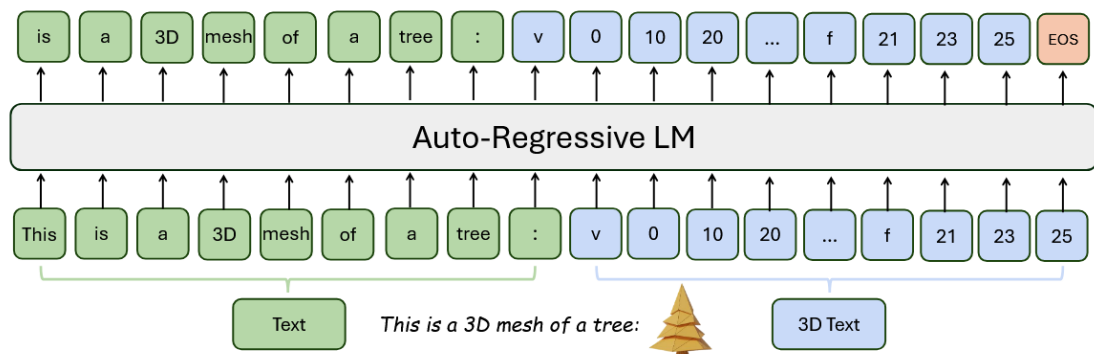


図 2.9: 3D メッシュをテキストトークンとして扱う 3D 生成システム LLaMa-Mesh[16]

MeshLLM[17] は LLaMa-Mesh で問題だった学習効率を, プリミティブメッシュを用いることで解決し, 分割したプリミティブメッシュから元の形状を再構成する学習プロセスにより, 生成品質とメッシュの意味を説明する能力を向上させた。

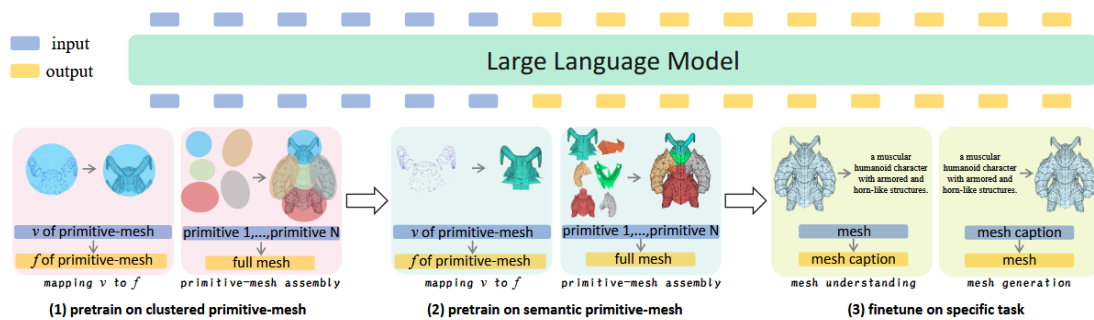


図 2.10: プリミティブメッシュを用いた 3D 生成システム MeshLLM[17]

3D-LLM[18] は 3D 点群データを LLM に学習させるというアプローチで 3D 空間の理解を目指し, 3D 空間のキャプション生成や質問応答を可能にした。

ShapeLLM[19] は 3D 点群から多視点の 2D 画像を蒸留して学習した「Recon++」と呼ばれるエンコーダによって 3D 空間の理解力を向上させた。

ShapeLLM-Omni[20] は 3D ボクセルを入力し, トークン化することで言語や視覚と同様に表現する。これによって 3D 生成や説明だけでなく, テキストによる 3D メッシュ編集を可能にした。



図 2.11: 3D 点群データを LLM で学習した 3D 生成システム 3D-LLM[18]

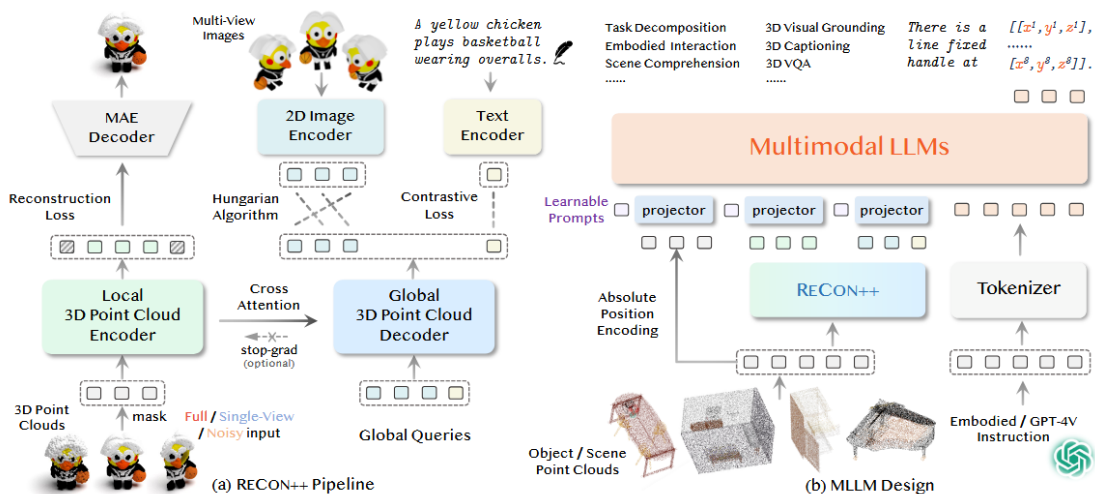


図 2.12: 多視点画像を学習した 3D 生成システム ShapeLLM[19]

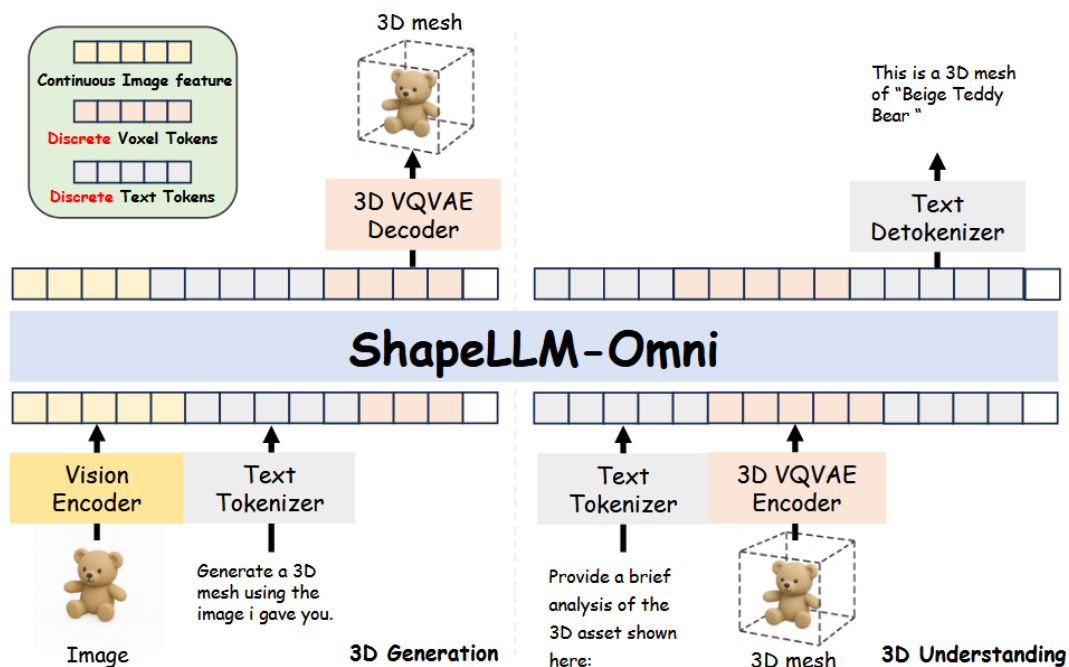


図 2.13: 3D ボクセルを学習した 3D 生成システム ShapeLLM-Omni[20]

2.3 既存のロボット設計手法

背景で述べた通り，既存のロボット設計手法は 3DCAD を用いる古典的な手法とある程度の役割を持ったモジュールどうしを接続してロボットを設計する手法に分かれている．特に後者の手法によって幅広いユーザがロボット設計を行うことが可能になった．例えば，Interactive Robogami[4] は足，胴，体などのパーツを並べてロボットを設計し，動力的な解析をフィードバックとして設計を支援する．パーツは展開図の形で出力され，折り紙のように折りたたんで組み立ててロボットを製作できる．

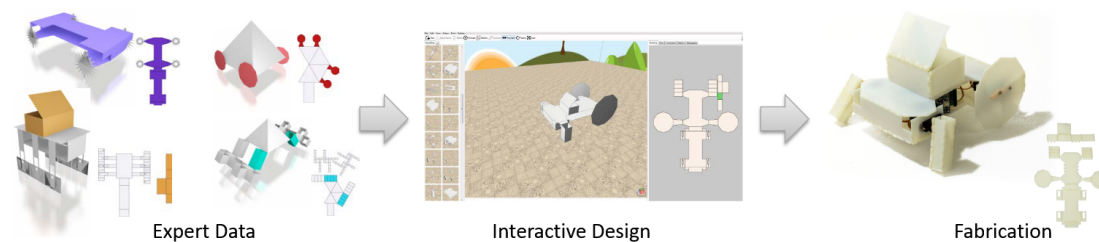


図 2.14: 折り紙モジュール型ロボット設計システム Interactive Robogami[4]

Interactive Robogami は折り紙構造を用いることで製作を簡略化したが，Megaro ら [21] は 3D プリントによって製作を簡略化した．この研究はユーザがロボットの構造を入力し，各リンクに一つの 3D モデルを割り当てて，ロボットの形状と構造を設計する．その後，設計したロボットで歩行シミュレーションを行い，安定して歩行できるように脚の太さや取付位置を変更し，歩行可能なロボット設計にする．そして，完成した設計を 3D プリンタで印刷して実世界でも歩行可能であることを確認した．

形状や構造ではなく，作業環境やタスクの入力からロボットを設計する手法も研究されている．この方法は入力した条件下で作業を行うための最適化を行うことで実用的なロボットを設計することができる．例えば RoboGrammar[22] は「胴体には脚が付く」というようなロボット構造を表現するためのグラフ文法を定義し，それに基づいて設計の初期案を作成した後，生成した構造のロボットで入力された地形上を歩行するシミュレーションと歩行の評価，構造の変更，行動の最適化を繰り返して強化学習を行い，最終的に入力された地形を踏破できるロボット構造を生成するシステムである．

RoboGrammar は，生物進化アルゴリズムに基づいて，リンクの長さの変更などを行う変異，歩行による移動距離の低い個体を淘汰する選択，既存の個体の特徴を組み合わせる新たな個体を生み出す交配などによりロボット設計を進化させてきたが，この方法はランダムな設計が生み出されるため，計算コストが高く，最適化に時間がかかった．この問題を解決するべく，ロボット設計の進化に LLM を導入したのが RoboMorph[23] である．RoboMorph はロボット設計と評価の結果

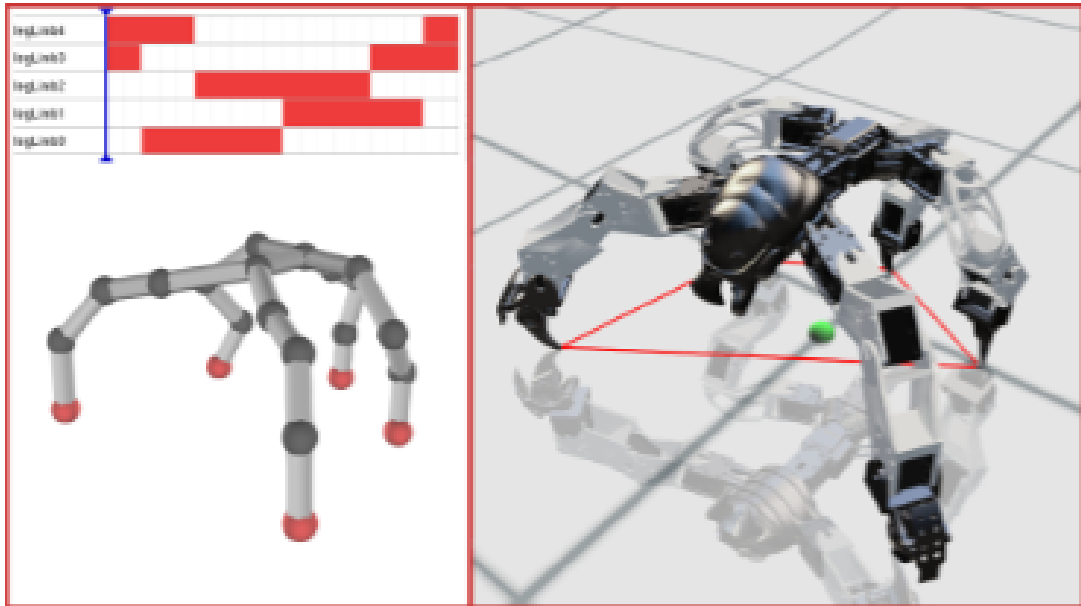


図 2.15: 3D プリント可能なロボット設計システム Interactive design of 3d-printable robotic creatures[21]

を LLM に入力し、設計の変更案を出力することで RoboGrammar に比べて効率的にロボット設計を進化させることに成功した。

さらに、生成 AI によってロボット設計を行う研究も進んでいる。RoboGen[24] は人間による設計をボトルネックとみなし、タスクの提案、環境の構築、学習を物理エンジン Genesis[25] によって行うことで人間抜きでのロボット設計を実現した。

進化的アルゴリズムにおいて、選択の際に重要になるのは報酬関数である。その報酬関数を LLM によって最適化したのが Eureka[26] である。Eureka は LLM にシミュレータのソースコードを入力し、ランダムな報酬関数から最適化を行い、人間の決めた報酬関数を超えるタスク成功率を示した。

Text2Reward[27] も同様にテキストで与えられたタスクを LLM がサブタスクに分割し、報酬関数を設計して強化学習を繰り返し行って人間よりも優れた報酬関数を設計した。ロボット設計は形状の入力から設計するものがほとんどだったが、構造の入力やタスク、環境の入力から初期の設計案を最適化していくことでロボット設計を行うことが増えてきた。

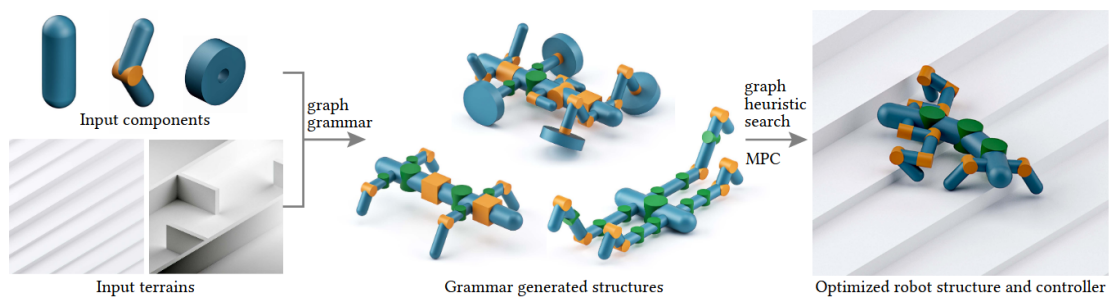


図 2.16: グラフ文法ロボット生成システム RoboGrammar[22]

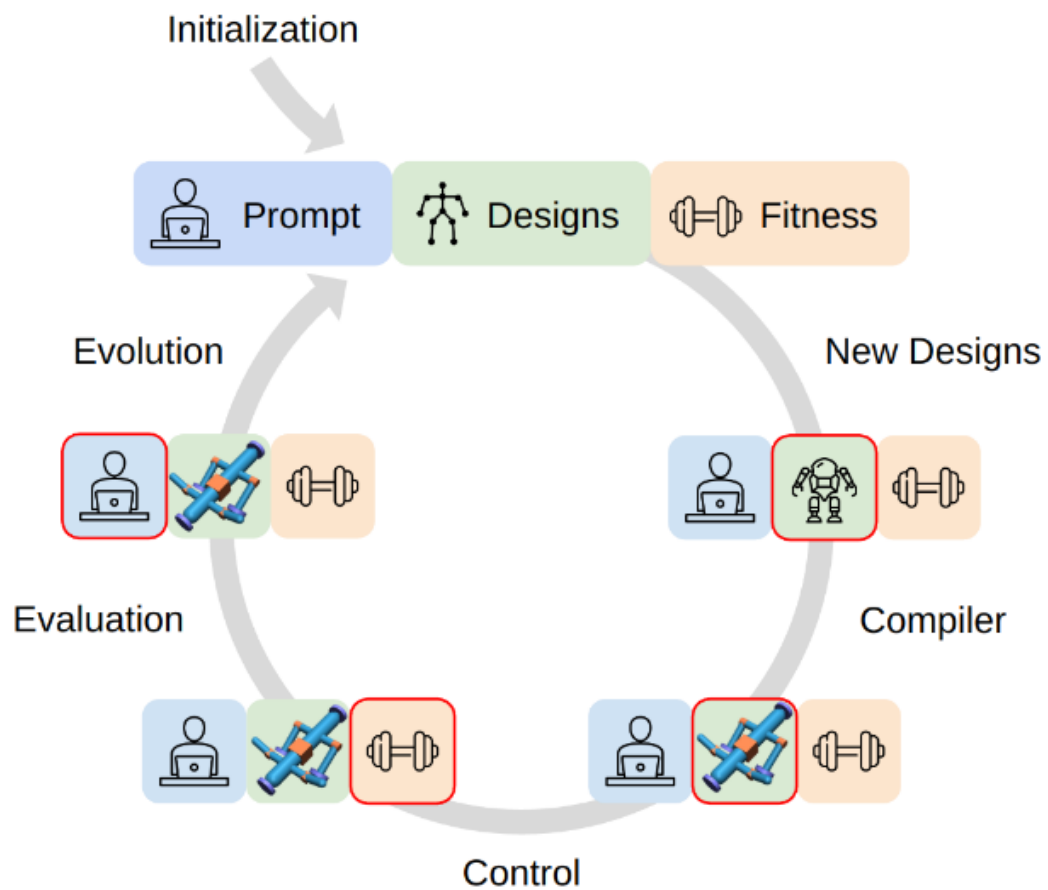


図 2.17: 進化的アルゴリズムによるロボット最適化システム RoboMorph[23]

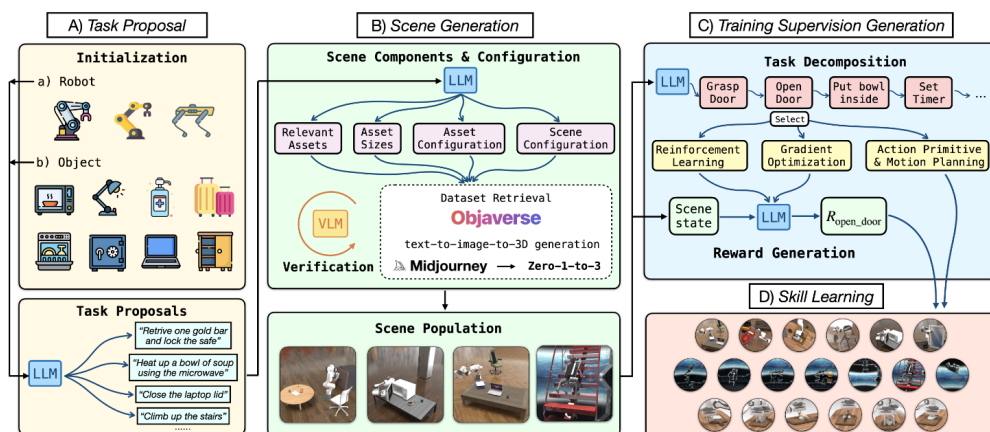


図 2.18: ロボット自動生成システム RoboGen[24]

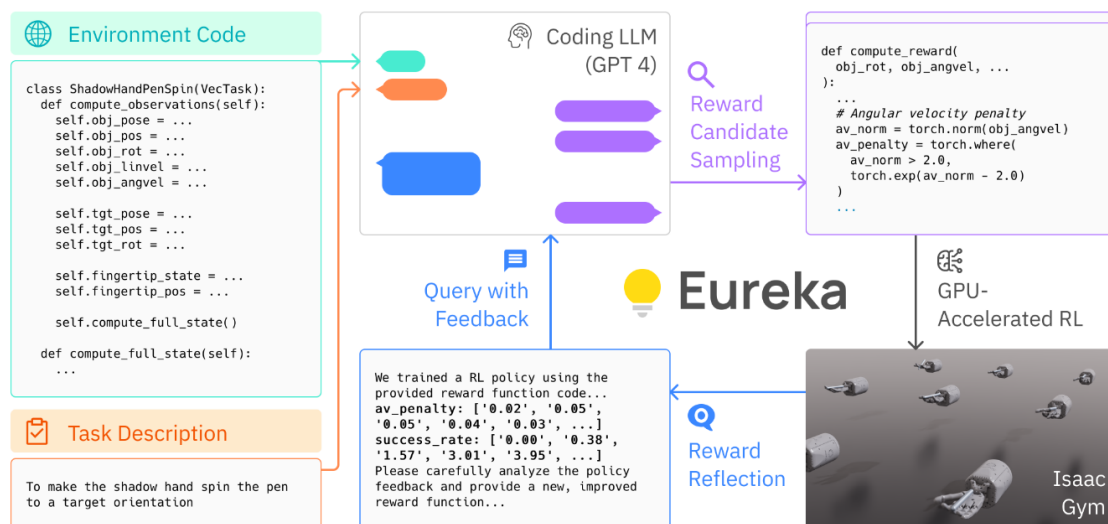


図 2.19: LLM によって報酬関数を決定するロボット設計システム Eureka[26]

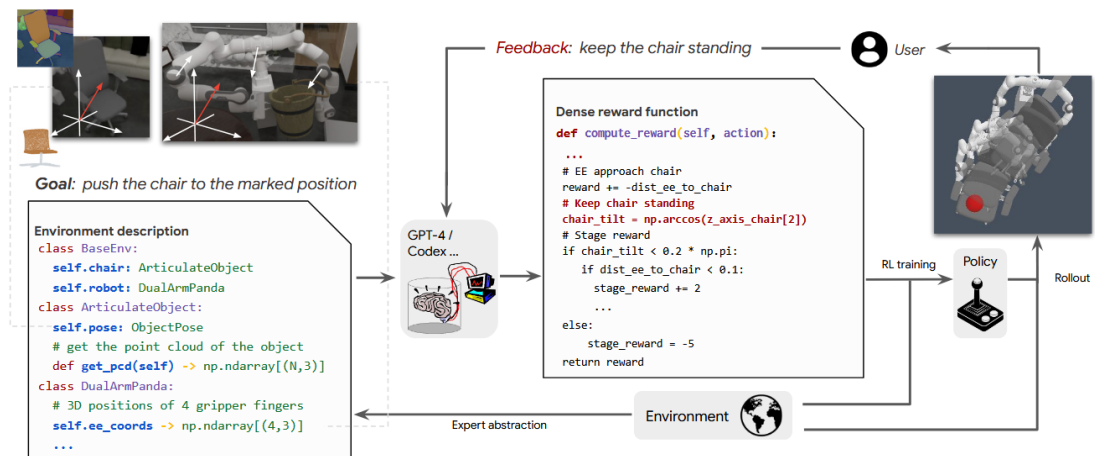


図 2.20: LLM がタスク分割と報酬関数を決定するロボット設計システム Text2Reward[27]

第3章 提案フレームワーク

本章では、開発した二つのシステム Sketch2Joints と Sketch2Robpla の関係について述べた後、運動学シミュレーション、URDF、条件付き画像生成、3次元モデル生成などの本論文に必要となる知識について述べる。

3.1 全体フレームワーク

ロボット設計は構造設計と形状設計に分けることができる。構造設計は主にロボットの動作に関係し、形状設計はロボットの外観と質量特性に関係している。Sketch2Joints ではユーザは記号化した関節を配置してロボットの構造を先に定義し、関節どうしの接続経路からリンクを自動作成して形状を定義する。この手法だと、ユーザはリンクの太さや断面形状を設計することができない。そこで、Sketch2Robpla では生成 AI「TRELIS」によって形状自由度を大きく向上させた。さらに構造設計と形状設計の順番を入れ替え、構造設計中に形状を反映した運動学フィードバックを行うよう変更した。これにより、Sketch2Joints でフィードバック不足によりユーザが満足感を得られないという問題を解決を目指した。また、Sketch2Joints では2次元平面上に関節を配置することで構造を設計していたが、これは実際に製作するロボットが3次元の物体であるために差異があり、操作も単調で何を設計しているのかわかりにくかった。Sketch2Robpla では先に3Dモデルを生成していることから、Sketch2Joints のような関節位置の定義はディスプレイ上での操作だと奥行の間隔が指定しにくいという問題もあり、入力を点から線分へと次元を増やすことによってユーザの操作から一意に関節が定義できるよう変更した。線分によるユーザ入力 Sketch2Joints では各リンクを3Dモデル化しているために必要が無かったモデルの分割も兼ねることができた。

3.2 運動学シミュレーション

ロボットにおいて関節の変位から関節の位置を求める計算を順運動学計算と言い、反対に関節の位置から関節の変位を求める計算を逆運動学計算と言う。ロボットの動きをイメージしたり、理解する上で重要な計算である。順運動学計算は3次元座標変換やロドリゲスの回転公式によって求められる。ここで、本研究で行う各関節の運動学計算から全体のロボット姿勢を計算する分散型運動学計算法の

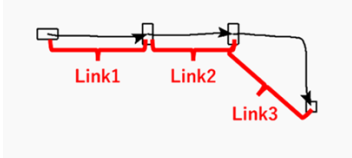
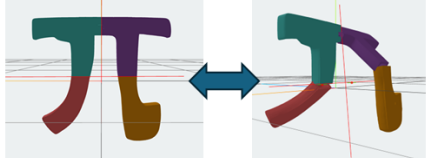
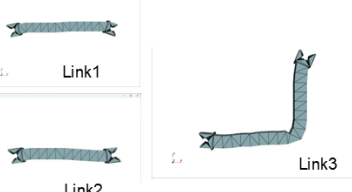
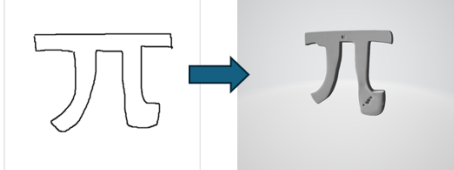
	Sketch2Joints	Sketch2Robpla
構造設計	記号化した関節配置(ユーザ入力) 	線分によるモデルの分割, 断面への関節配置 
形状設計	入力された関節構造からリンクを自動作成 	スケッチから生成AIによる3Dモデルを生成 

図 3.1: 提案システム Sketch2Joints と Sketch2Robpla の比較

手順を説明する．まず，回転した関節 J_0 とその下位関節 J_1 について考える． J_0 の位置ベクトルを $\mathbf{r}_0$ とし， J_1 の位置ベクトルを $\mathbf{r}_1$ とする．この時，関節間の距離を (1) 式で求める．

$$\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_0 \quad (3.1)$$

次に，回転した関節の回転軸ベクトルを $\mathbf{n}$ ，回転した角度を θ としてロドリゲスの任意軸周りの回転公式に基づく (2) 式に代入することで， J_0 から見た回転後の J_1 の位置ベクトル $\mathbf{r}'$ を求める．

$$\mathbf{r}' = \mathbf{r} * \cos \theta + (1 - \cos \theta)(\mathbf{r} \cdot \mathbf{n}) + (\mathbf{n} \times \mathbf{r}) * \sin \theta \quad (3.2)$$

この $\mathbf{r}'$ に $\mathbf{r}_0$ を足すことで回転後の J_1 の位置ベクトルを求める．CAD でロボット設計をしている場合は，部品どうしをアセンブリしてロボットの動きを確認し，部品を修正していた．逆運動学計算はヤコビアンを用いて解析的に求めていたが，近年ではニューラルネットワークを用いて計算する手法がほとんどである．本研究では，UI 上のスライダーで関節の変位を操作し，ロドリゲスの回転公式によってシステムが順運動学計算を行ってロボットの動作をユーザに対してフィードバックした．

3.3 URDF(Unified Robot Description Format)

URDF ファイルはその名の通り、ロボットの情報を記述するためのテキストファイルである。マークアップ言語であるため、文法は HTML(Hyper Text Markup Language) と類似している。URDF ファイルの例を図 3.2 に示す。Link, Joint といったタグの中にリンク名、関節位置、関節の軸、接続関係などの情報を階層構造で記述することができる。URDF はロボット設計の受け渡しに使われ、ROS(Robot

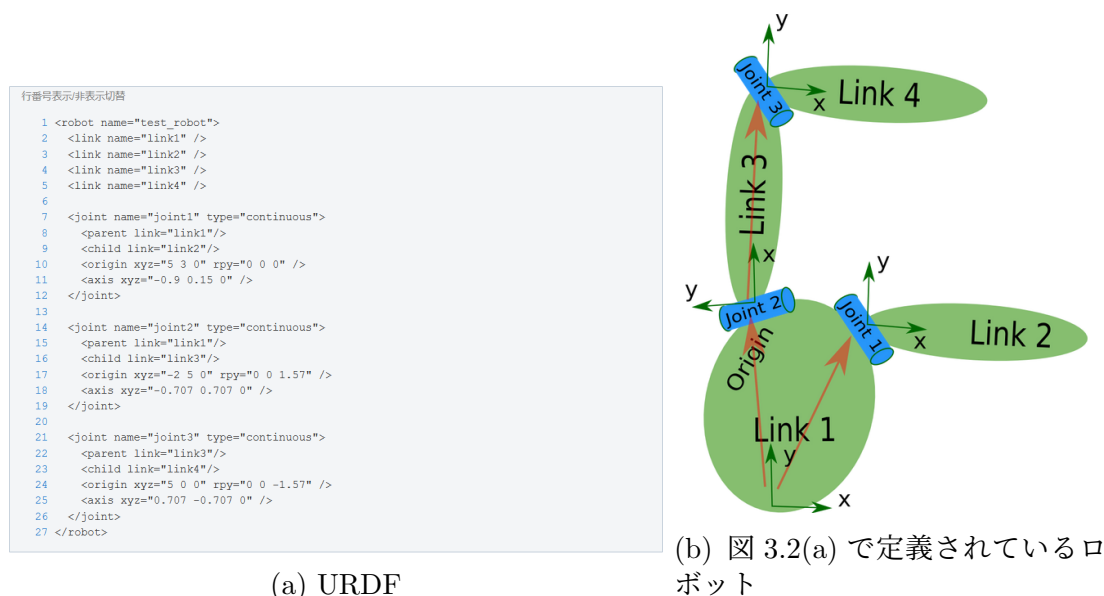


図 3.2: URDF の例

Operation System) や Issac Sim など様々なシミュレータに対応している。近年、URDF のデータセット [28] も作成され、ロボティクス分野において重要なファイル形式である。

3.4 条件付き画像生成

本研究ではユーザのスケッチ入力から ControlNet[29] と Stable Diffusion[30] を用いて画像を生成する。ControlNet が抽出したスケッチの輪郭情報、深度情報、構造情報を抽出し、情報の詳細さで分類された多階層の特徴マップに変換して Stable Diffusion のデコーダーに入力する。Stable Diffusion はテキストの特徴量と ControlNet の特徴量を合わせて、デコーダーによって画像を出力する。つまり、ControlNet は構図、ポーズ、輪郭線などの構造的な条件を追加して制御する役割を持ち、Stable Diffusion はテキストと ControlNet に制御される形で画像を生成している。本研究ではこの生成手法を、ユーザのスケッチを画像に反映するために使用している。

3.5 3次元モデル生成

本研究ではユーザのスケッチを元に生成した画像から 3D モデルを生成する際に TRELLIS[9] を用いる。TRELLIS は Microsoft の開発した AI モデルであり、テキストや画像から 3D モデルを生成することができる。TRELLIS の特徴は SLAT (Structured LATents: 構造化潜在表現) と呼ばれる表現である。SLAT は疎なグリッドの交点に形状、色、質感などの情報を持つトークンを配置する表現である。この SLAT を生成するために、TRELLIS は Rectified Flow と呼ばれる生成モデルを用いる。Rectified Flow は初期のランダムノイズから 2D 画像やテキストで示された形状に少しずつ変化させていき、完全な SLAT を生成する。これにより、3D モデルを様々な視点で見た際の一貫性の向上や特定の部位の編集が可能になり、多用途で高品質な 3D 生成を実現した。

第4章 Sketch2Joints

Sketch2Joints はロボット設計の民主化を目指し，ロボットの関節，リンクを記号化することでロボットに関する知識の無いユーザにも関節構造の設計を可能にしたインターフェースである．ユーザ実験によって運動学フィードバックの必要性，ロボット構造の記号化によってロボット設計が非専門家にも理解しやすいものになったかを確認した．本章では Sketch2Joints の手法，結果について述べる．

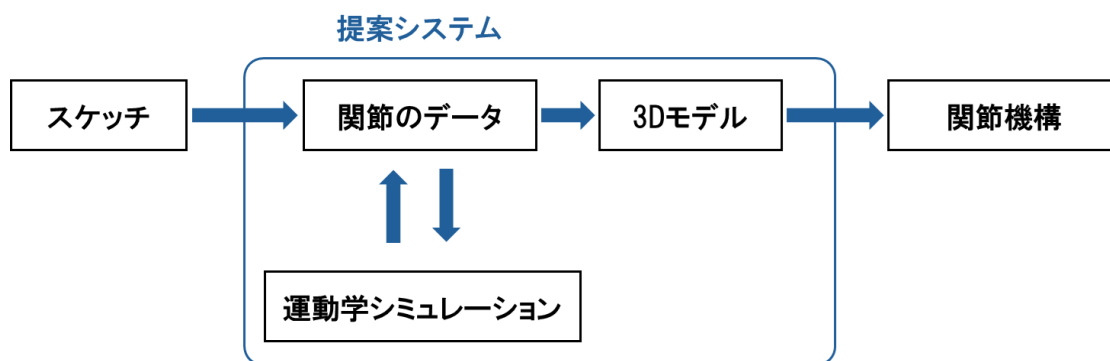


図 4.1: 提案システム Sketch2Joints 概要図

4.1 手法

4.1.1 設計インターフェース

Sketch2Joints のインターフェースのスケッチ中の画面を図 4.2 に示す．赤枠内はスケッチによってロボットアームの設計を行う領域である．青枠内はロボットアームの新規作成や保存を行う領域である．緑枠はスケッチを行う際の特有のメニューフレームで，スケッチモードの切り替えや各関節の編集を行う．スケッチモードは Point, Relation, Delete の 3 種類あり，それぞれ関節の配置，リンクの作成，関節の削除を行うモードであり，Point モードで既に配置した関節をクリックすると，図 4.3 に示す関節編集フレームが緑枠の SketchMenu フレーム下部に表示される．このフレーム内では関節のタイプ，回転軸ベクトル，関節位置ベクトルの Z 成分を編集することができる．関節の回転軸ベクトルを変更すると，値に応

じて異なる表示がされる．例えば，X 軸回転の関節は縦向きの長方形，Y 軸回転の関節は横向きの長方形，Z 軸回転の関節は円で表示される．Relation モードに切り替えると，SketchMenu フレームにリンクの種類を選択できるセレクトボックスが表示される．このセレクトボックスの表示が Linear の場合にはリンクの経路は直線で与えられ，Free の場合は，図 4.2 に示すスケッチのようにユーザが自由にリンクの経路を決めることができる．ユーザのストロークを等距離で分割し，分割点を制御点としたスプライン曲線をスケッチ上に表示している．ユーザがスケッチを変更する度に，変更内容は以下に示す関節データに反映される．

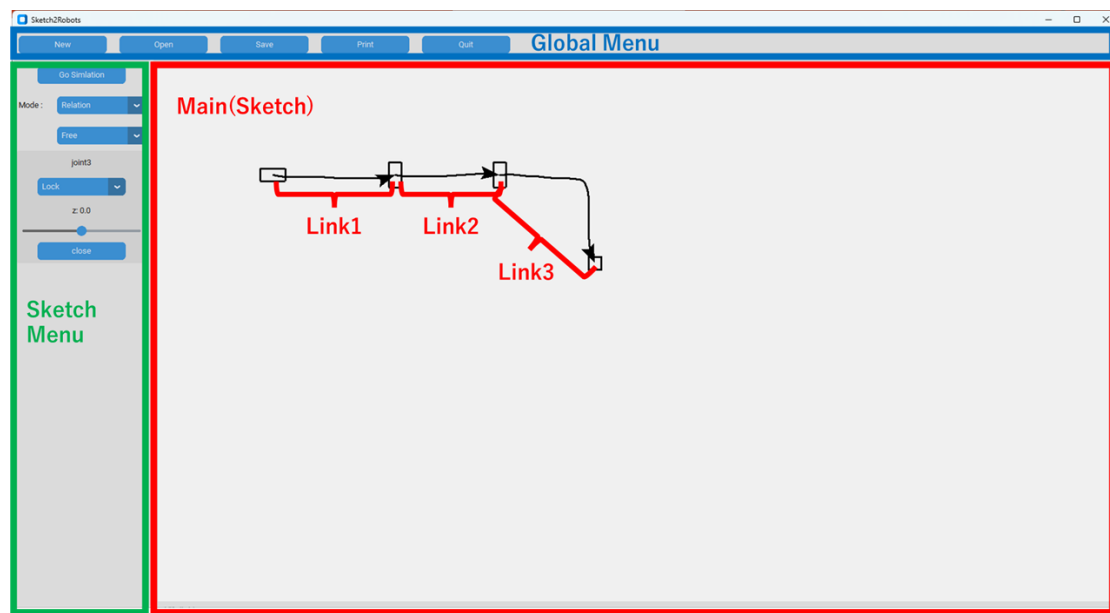


図 4.2: 提案システム Sketch2Joints のインターフェース

1. 関節 ID
プログラム内で関節を認識するための ID
2. 位置ベクトル
関節の位置座標を表す 3 次元のベクトル
3. 移動軸ベクトル
関節の回転軸を表す 3 次元のベクトル，直動軸の場合は移動方向を表す
4. 稼働範囲
回転や直動の上限と下限を表す二つの値
5. 上位関節のリスト
関節の上位 (親側) の関節の ID のリスト，原則一つしか存在しないがリンク機構の場合は例外であり，複数の関節のリストになる．

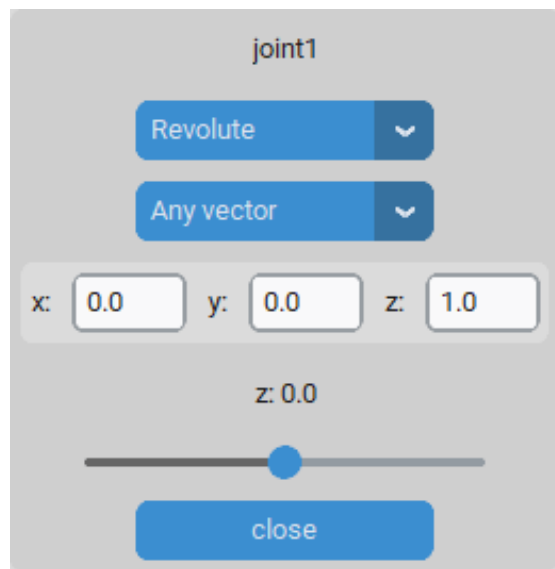


図 4.3: 関節編集フレーム

6. 下位関節のリスト

関節の下位 (子側) の関節の ID のリスト, このリスト内の関節は回転や直動の影響を受ける.

7. 関節のタイプ

関節の変位が変わったとき, どのような動きをするかの種類, 回転 (Rotate), 直動 (Linear), 固定 (Fixed) がある.

4.1.2 ユーザフィードバック

提案システムのシミュレーション画面を図 4.4 に示す. 緑枠内は図 4.5, 4.6 に示す Motor タブ, Axis タブが表示される. スケッチが存在する, もしくは事前にスケッチ等から作成したロボットアームのデータが読み込まれている場合, Motor タブに可動関節の数だけスライダが表示される. このスライダの値は各関節の変位を表しており, 値に応じて運動学計算を行い赤枠に表示されるシミュレーション結果が変更される. Axis タブを表示することでグラフの軸の表示範囲を編集することが可能になり, シミュレーションフレームの中心にロボットアームを表示することができる.

4.1.3 関節の製作

ユーザが作成した関節のデータから, 関節位置, 関節種類を反映して, 3D プリントで印刷可能な関節間をつなぐリンクの 3D モデルを作成する. ロボットアーム

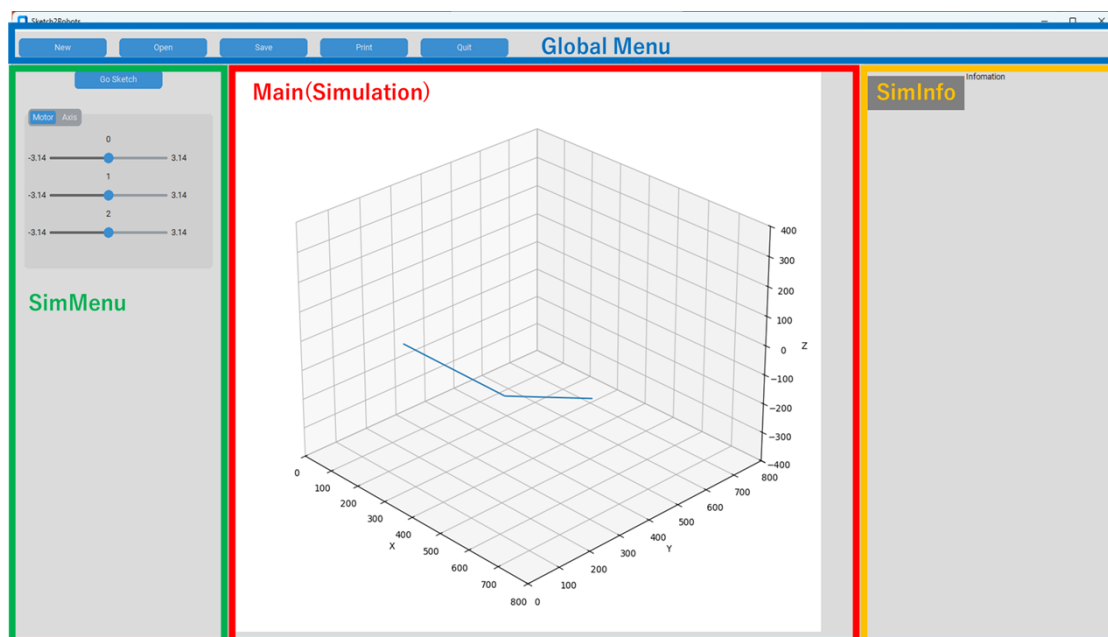


図 4.4: シミュレーションインターフェース

のアクチュエータには Robotis 社 Dynamixel XL330-M077(以下 XL330) を利用した。アクチュエータを取り付ける部品は予め設計している。回転機構の回転軸の向きに応じて図 4.7(a),4.7(b),4.7(c) のいずれかにアクチュエータをはめ込み、図 4.7(d) を軸側に取り付けることで、関節モジュールとして完成する。また、ユーザがスケッチした経路を基に関節モジュール間をつなぐリンクを作成する。図 4.2 のスケッチから作成したリンクの 3D モデルを図 4.8(a),4.8(b),4.8(c) に示す。各関節モジュールと、リンクの接続部は同じ形状になっており、宮大工の技法である宮島継ぎにより部品間を接続するとロボットアームが完成する。宮島継ぎは部品 A と部品 B の同じ形の接続部を重ね合わせ、中心に円柱形の軸を入れることで固定する継手であり、ねじ止め箇所が減るため組立、組み換えが容易になる。製作には QIDI TECH Q1 PRO と FlashForge Adventurer 5M を使用した。製作時間は素材が PLA の場合、約 3 時間、ABS の場合、約 5 時間。重量は PLA の場合、Link1,2,3 がそれぞれ 31 グラム、27 グラム、40 グラム、ABS の場合、Link1,2,3 がそれぞれ 48 グラム、41 グラム、61 グラムだった。

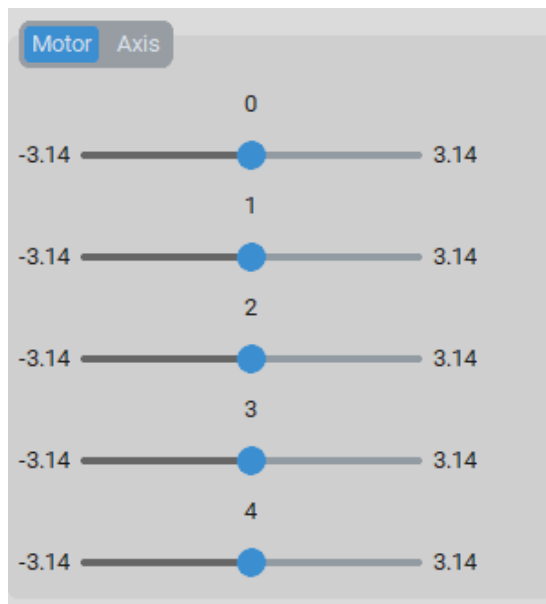


図 4.5: Motor タブ

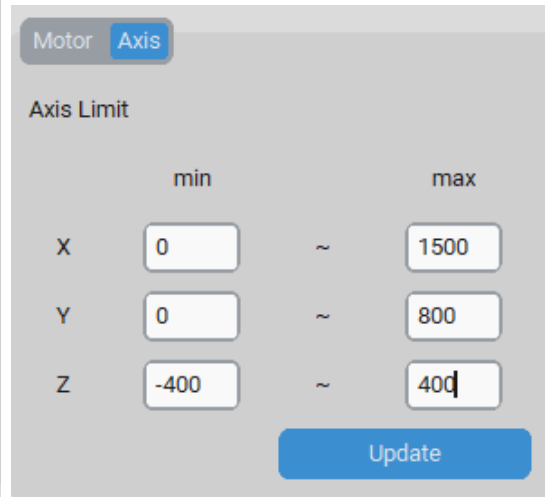
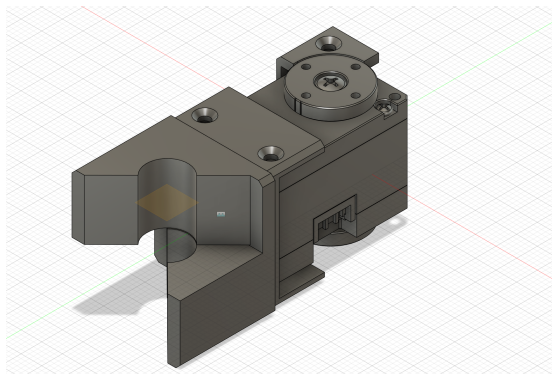
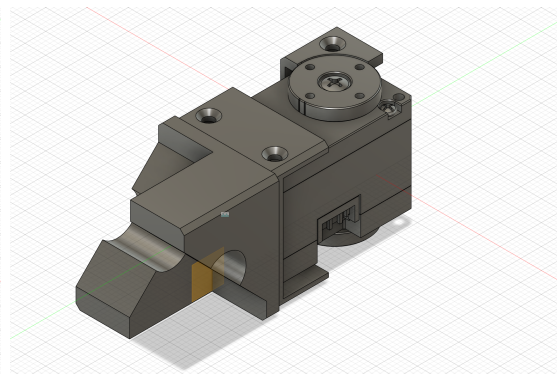


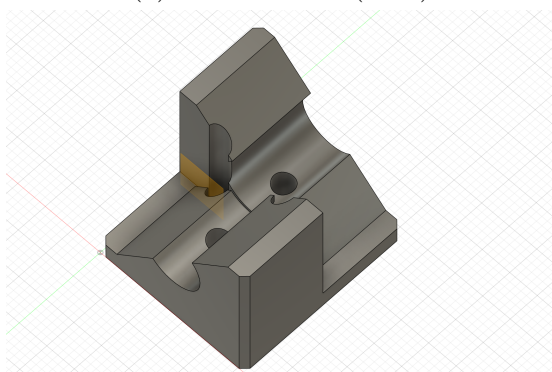
図 4.6: Axis タブ



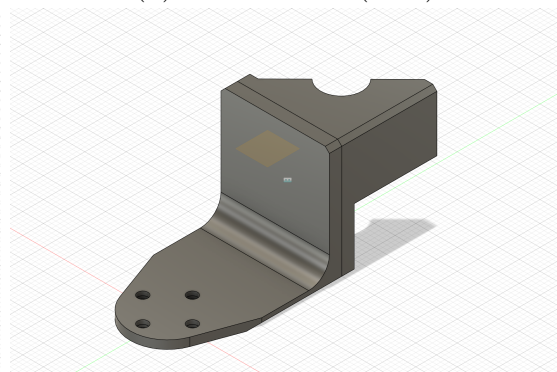
(a) 垂直回転機構 (X 軸)



(b) 垂直回転機構 (Y 軸)

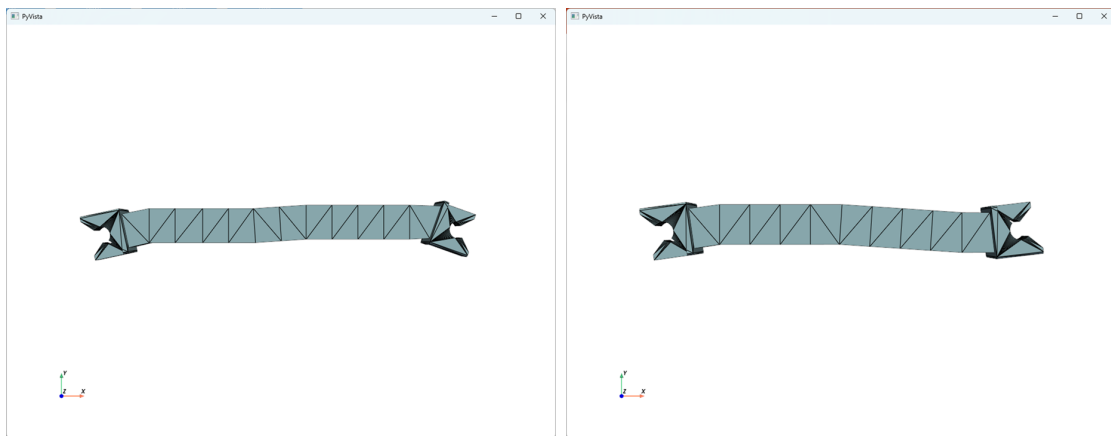


(c) 水平回転機構 (Z 軸)



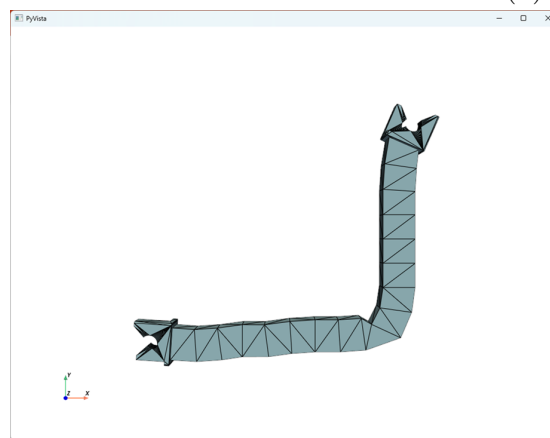
(d) 軸側取付部品

図 4.7: 関節



(a) Link1

(b) Link2



(c) Link3

図 4.8: 図 4.2 のスケッチから作成したリンク

4.2 結果

4.2.1 製作

関節機構

垂直回転関節，水平回転関節を図 4.9(a),4.9(b) に示す．両端が宮島継ぎの形状をしているため，他関節やリンクと接続することができる．製作に使用したプリンタは QIDI TECH Q1 PRO で ABS を使用した．製作時間は，垂直回転関節，水平回転関節共に約 1 時間であった．



(a) 垂直回転関節



(b) 水平回転関節

まず，6 自由度の冗長ロボットアームを図 4.10(a) のスケッチで設計した．スケッチ自体はほぼ一直線に関節を配置し，接続関係で順に繋いだ．関節の種類は X,Y,Z の 3 種類を交互に配置し，前半のリンクと後半のリンクでは前半が長くなるように設計した．その際のシミュレーション画面を図 4.10(b),4.10(c) に示す．シミュレーション画面で動作を確かめた後，図 4.10(d) に示す 3D モデルを出力した．スケッチ上の距離に応じて，長さの異なるリンクが出力されていることが分かる．次に，ぬいぐるみの画像をスケッチウィンドウに読み込み，図 4.11(a) に示すぬいぐるみの内部構造のスケッチを作成した．このスケッチから作成した関節機構のシミュレーションの画面を図 4.11(b) に示す．スケッチ時に入力した位置関係とシミュレーションが一致していることが確認できる．また，同じデータから作成したリンクを図 4.11(c) に示す．これも様々な長さのリンクが出力されていることがわかるが，直線状のリンクのみが出力されており，リンクの分岐に対応できていない．分岐を行うモデルを別途作成する必要がある．

制御方法

制御には 2 枚の基板を使用した．図 4.12(a) は PC から指令を受け取り，XL330 に信号を送信する基盤である．USB type-c ケーブルを用いて PC と接続し，3 ピン

の JST コネクタケーブルを用いて図 4.12(b) に示す DXSharingBoard と接続する。DXSharingBoard は XL330 への電源供給を行う。また、複数のモータとの接続が可能な基盤である。実際の接続例を図 4.12(c) に示す。また、信号は ROBOTIS 社のソフトウェア、Dynamixel Wizard から送信する。

4.2.2 ユーザスタディ

提案システム Sketch2Joints をユーザに体験してもらい、設計したロボットアームを 3D プリンタを用いて製作した。製作には Adventurer 5M と Bambu Lab X1 Carbon の 2 台の 3D プリンタを用い、材質は PLA で製作した。SUS による定量的な評価とインタビューによる定性的な評価を行い、Sketch2Joints のユーザビリティとユーザ満足度を確かめた。

実験方法

Sketch2Joints の評価はユーザ実験によって行った。20 代の男女 4 名の被験者は、本システムを用いて図 4.11(a) でスケッチに用いたものと同じぬいぐるみの内部構造を設計した。UI の直観性を示すため、操作方法の詳しい説明は行わなかった。被験者は開始 1 分程度で関節を配置し始め、その後すぐに関節間の接続関係を追加し始めた。10～20 分程度で被験者たちは自分の満足するスケッチを作成した。作成後、被験者に System Usability Scale(SUS) に基づくアンケートを行った。また、作成したスケッチを基にロボットを作成し、ロボットの達成度や満足度に関するアンケートを行った。

定性的評価

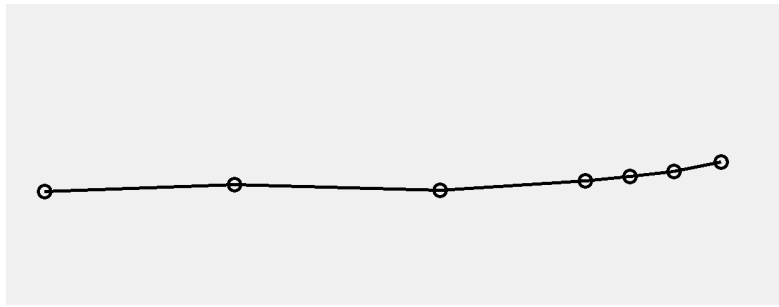
被験者のスケッチを図 4.13 に示す。図 4.13(a),(b) は Sketch2Joints でロボットアームとしてシミュレーションと製作が可能だが、図 4.13(c) は構造が 2 つに分かれてしまっており、2 つのロボットアームとして入力される。図 4.13(d) は関節で閉ループができており、リンク構造として入力される。この 2 つのスケッチはシミュレーション時に不具合が発生してしまう。この問題は前節で挙げた UI が表示する情報の不足が原因である。図 4.13(a) のロボットアームの製作結果を図 4.14 に示す。3D プリンタを用いて製作した内部構造をぬいぐるみに挿入し、被験者がポーズをとらせた。部品は約 5 時間で製作できた。被験者に聞き取りをした結果、製作した内部構造はイメージ通りの動きができたことがわかった。

定量的評価

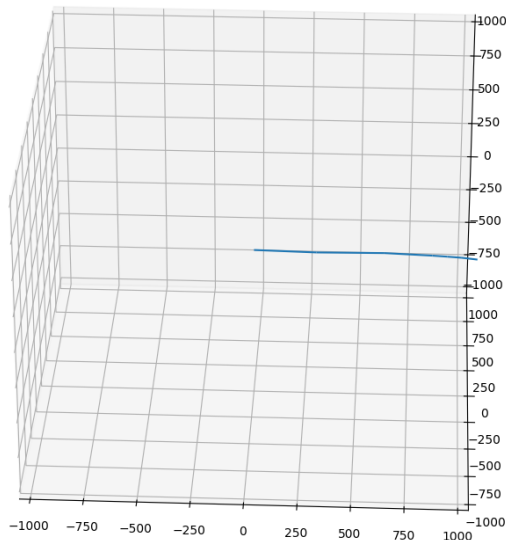
SUSに基づくアンケートの結果を図4.15に示す。平均スコアは65である。「このシステムを頻繁に使用したい」「このシステムは必要以上に複雑だと思う」「このシステムはスムーズに機能し、連携がとれていると思う」「このシステムには不規則な点が多いと思う」「ほとんどの人がこのシステムをすぐに習得できると思う」「このシステムを使い始める前に学ぶべきことはたくさんあると思う」に関しては被験者の意見が概ね一致しており評価も高い。しかし、「このシステムはシンプルで使いやすい」「このシステムは時間がかかると思う」の二つの質問についてはユーザの評価が少し別れた。だが、「このシステムは必要以上に複雑だと思う」「ほとんどの人がこのシステムをすぐに習得できると思う」については良い評価でまわっているため、この評価の差は被験者がイメージした構造の複雑さが関係している可能性がある。「このシステムを使うにはテクニカルサポートが必要」と「このシステムを使っていると自信が持てる」という2つの質問については良い評価を得られなかった。前者は情報不足、後者はフィードバック不足によるものだと考える。改善案として、関節の回転方向の表示や接続関係の矢印で表示して順序を表すなど、UI上の不足している情報を補うことと、スケッチの変更ごとにシミュレーション画面にプロットするなど、スケッチを変更した際のユーザへのフィードバックを明確にすることが挙げられる。

4.2.3 動作実験

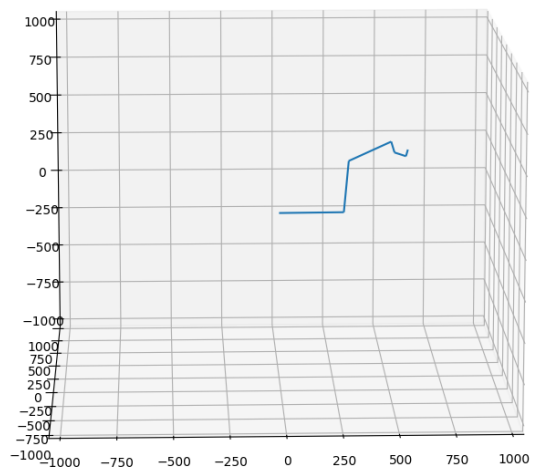
図4.16に示す製作したロボットアームを左下のスケッチと比べると、関節の種類と個数、リンクの形状が一致していることがわかる。床に置いた50グラム程度の物体をロボットアームで押して移動させる実験を行った。この実験の様子を図4.17に示す。想定通りに物体を移動させることができた。次に、アクチュエータによってロボットアーム自身を移動させる実験を行った。この実験の様子を図4.18に示す。先端の折れ曲がっているリンクが、床をひっかく様にモータを制御したが、自重を持ち上げるだけのトルクが出ず、移動させることはできなかった。この2つの実験から、設計したロボットアームが重力を受ける方向への動作しないということがわかった。停止の際、過積載のエラーが発生してモータが止まっているため、原因はモータのトルク不足と思われる。



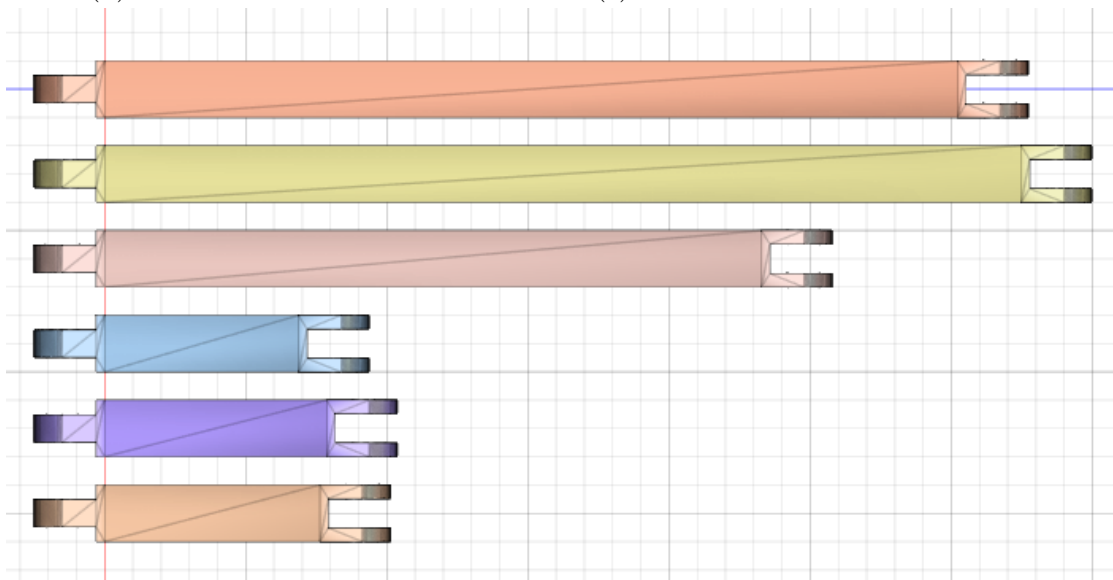
(a) スケッチ画面



(b) ロボットアームの初期姿勢

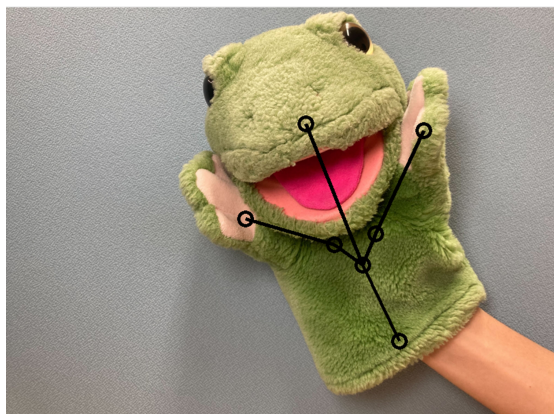


(c) 関節回転後のロボットアームの姿勢

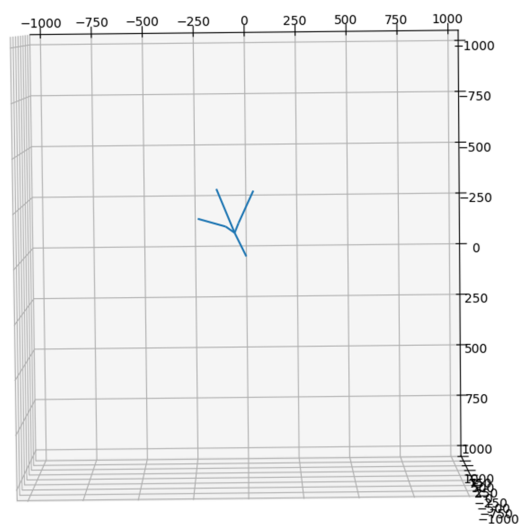


(d) 作成したリンクのデータ

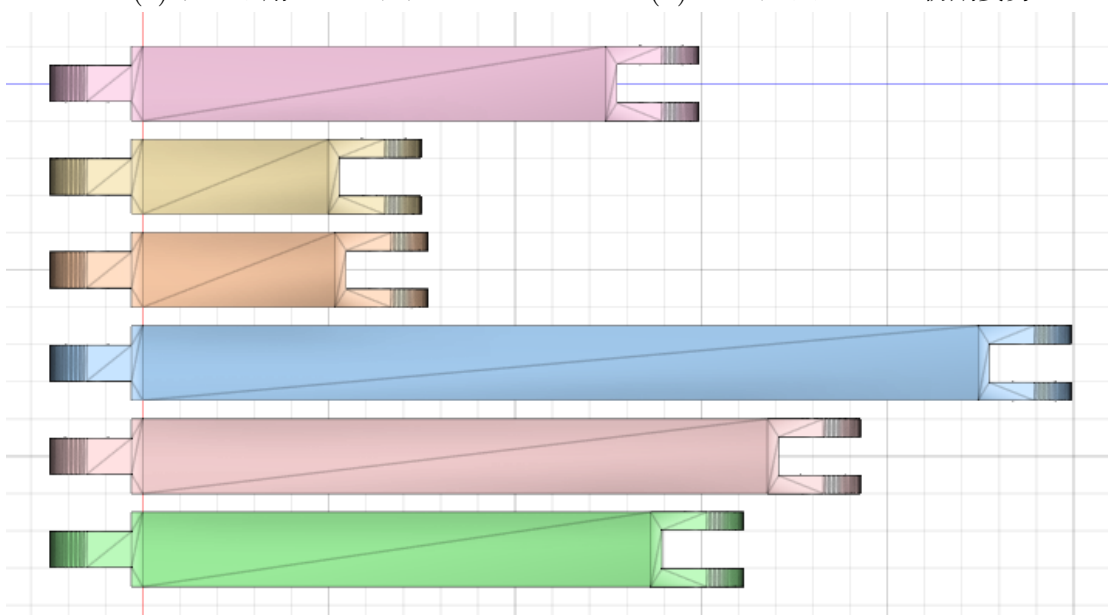
図 4.10: Sketch2Joints の入出力実験結果 1



(a) テスト用のスケッチ



(b) ロボットアームの初期姿勢

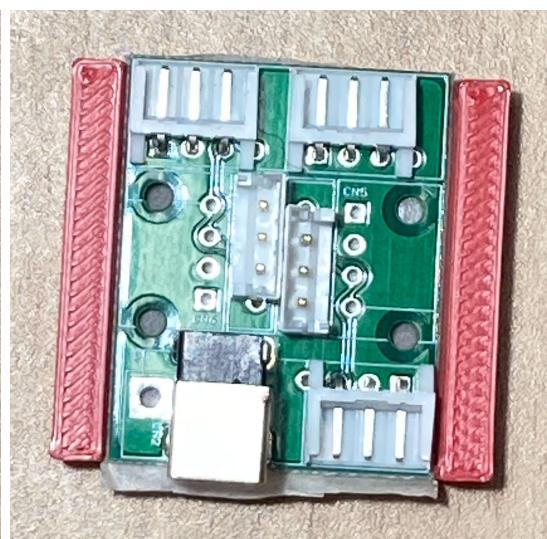


(c) 作成したリンクのデータ

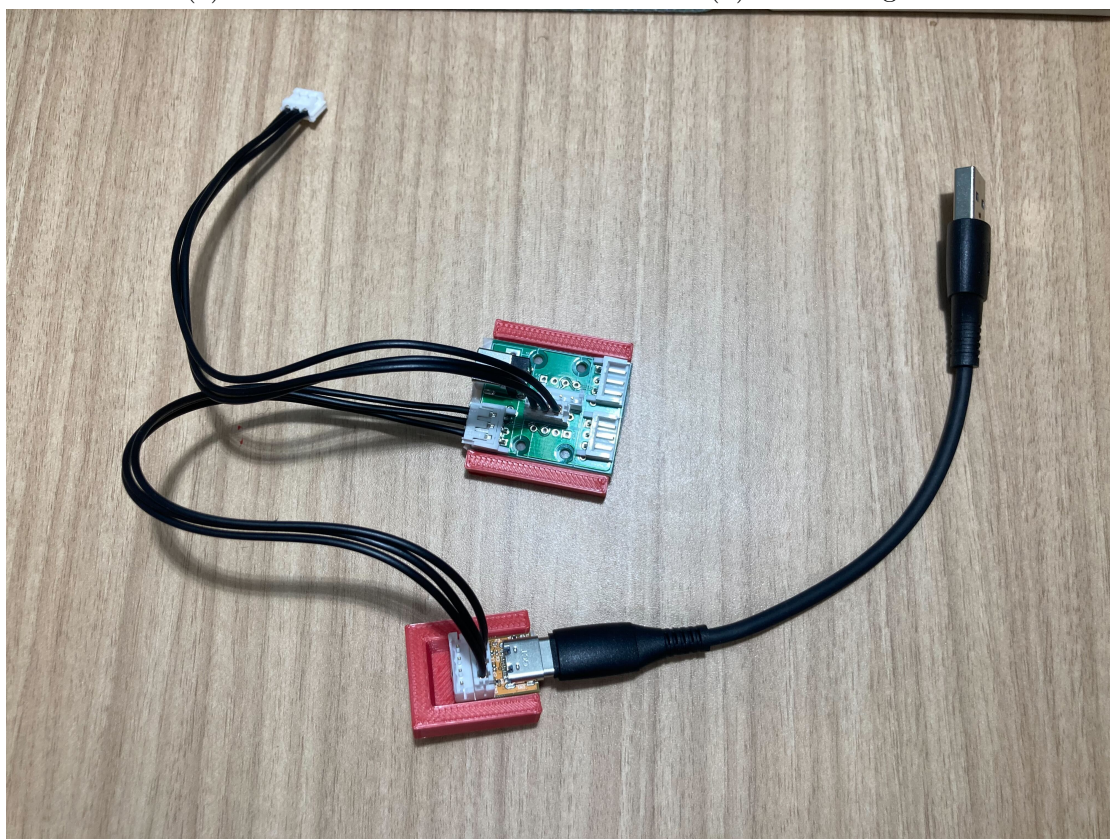
図 4.11: Sketch2Joints の入出力実験結果 2



(a) USB2DXIF

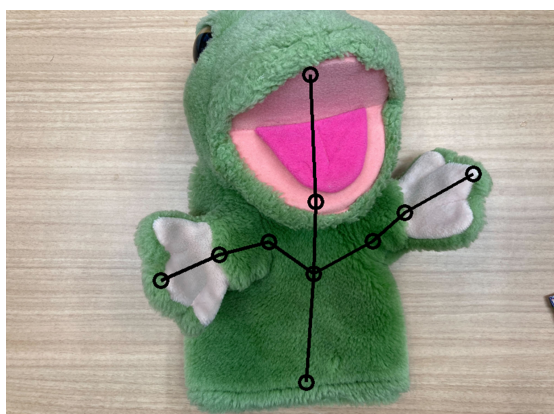


(b) DXSharingBoard

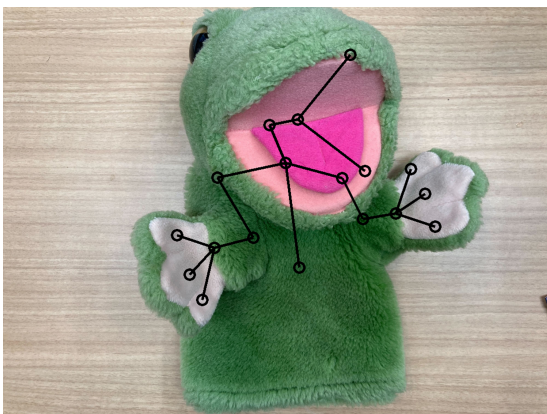


(c) 各要素を接続した様子

図 4.12: 製作したロボットアームの各要素



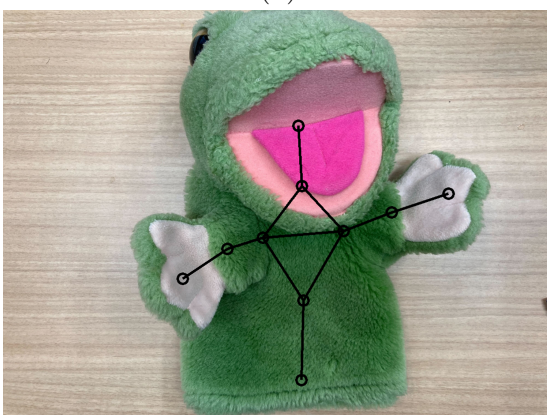
(a)



(b)



(c)



(d)

図 4.13: 被験者の入力した関節構造

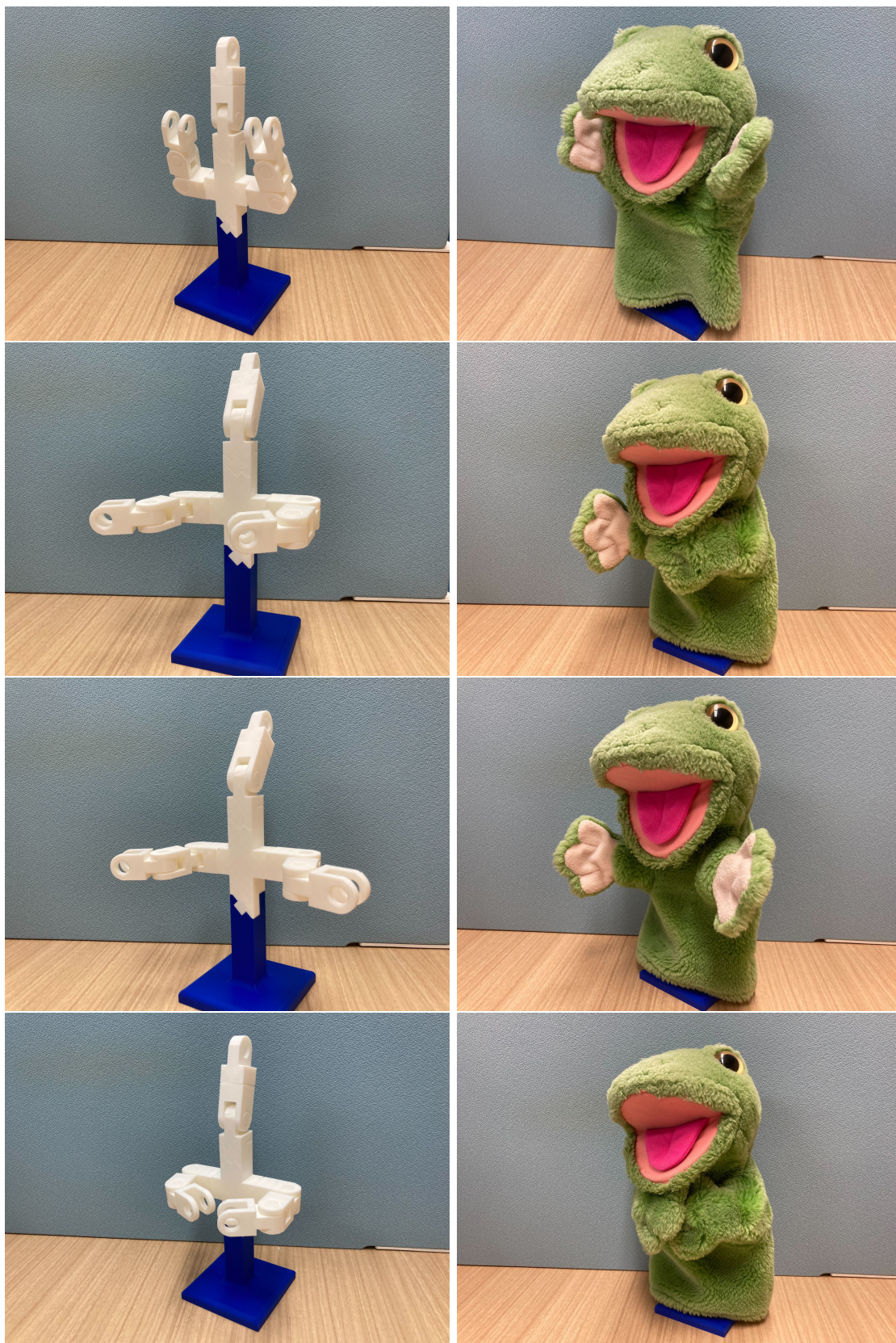


図 4.14: 内部構造 (左) と内部構造挿入後のぬいぐるみ (右)

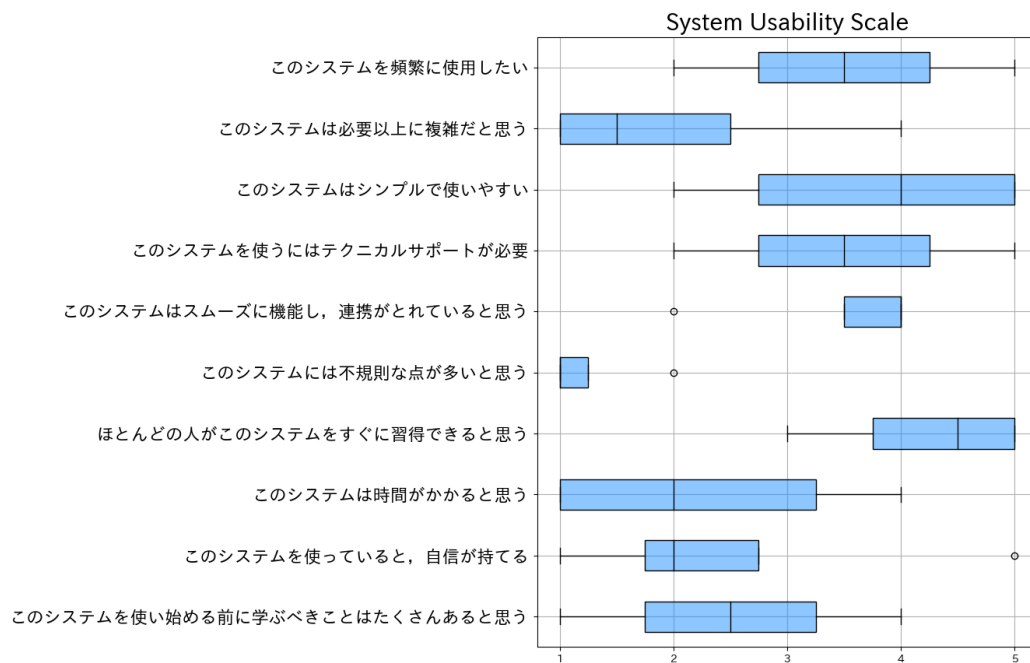
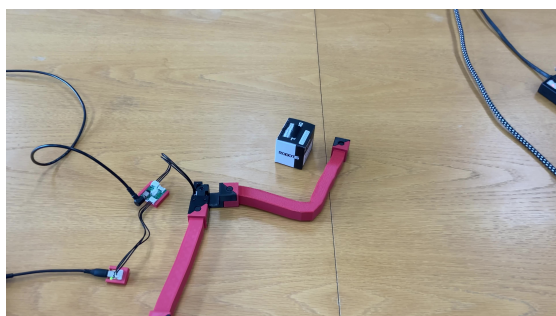


図 4.15: System Usability Scale に基づくアンケートの結果



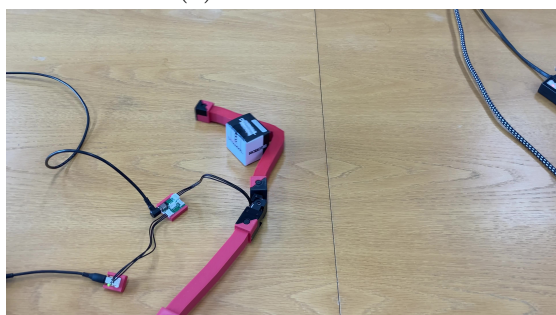
図 4.16: 図 4.2 のスケッチから作成したロボットアーム



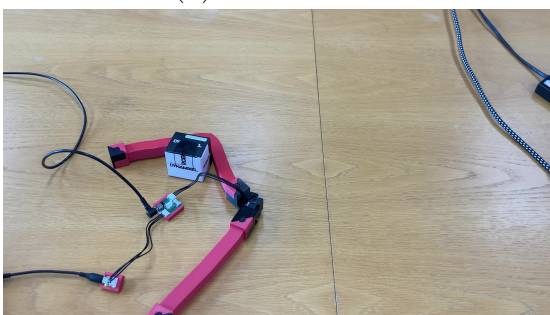
(a) 実験の様子 1



(b) 実験の様子 2



(c) 実験の様子 3



(d) 実験の様子 4

図 4.17: 物体を運ぶ実験の様子



(a) 実験の様子 1



(b) 実験の様子 2



(c) 実験の様子 3



(d) 実験の様子 4

図 4.18: ロボットアーム自身の移動実験の様子

第5章 Sketch2Robpla

現状のロボット設計は必要以上に複雑な 3DCAD と精密な数値計算によって行う手法と、予め用意されたモジュールの限定された設計空間内での探索を人間、もしくはアルゴリズムが行う手法の二つである。この二つは操作の簡単さと形状自由度がトレードオフの関係になっている問題があるため、スケッチインタフェースによる直感的な操作とスケッチからの 3D モデル生成によって高い形状自由度の設計を目指したシステム Sketch2Robpla を提案する。本章では、Sketch2Robpla の手法及び結果について述べる。

5.1 手法

Sketch2Robpla のパイプラインを図 5.1 に示す。Sketch2Robpla は形状生成インターフェースと構造定義インターフェースの 2 つのインターフェースで構成されている。ユーザのスケッチを形状生成インターフェースに入力し、3D モデルを生成、生成した 3D モデルを関節定義インターフェースに入力し、ユーザの操作によってモデルを分割して分割結果を基に関節を定義し、URDF(Unified Robot Description Format) ファイルを作成する。さらに 3D モデルに実動用の構造を付加し、印刷用のデータを出力する。Sketch2Robpla はユーザのスケッチから実世界で動くロボットとそのデータを作成する一気通貫のシステムである。

5.1.1 形状生成インターフェース

形状生成インターフェースを図 5.3 に示す。このインターフェースはユーザの想像しているロボット形状スケッチを入力とする。スケッチは古来より人間に表現手法として用いられており、子供のころにほとんどの人々が経験しているため、スケッチ入力を採用することで、多くの人々が直感的にロボット形状を入力することができる。そして、入力されたスケッチから画像を生成する。3D モデルを生成する前に画像生成を挟むことによって、スケッチに苦手意識を持っているユーザを安心させ、スケッチにかかる労力を軽減することができる。画像生成には ControlNet と Stable Diffusion を利用しており、生成時には図 5.4 に示すパラメータを調整することができる。例えば、プロンプトにロボットの種類や質感を入力すれば、生成される画像の品質は向上する。スケッチよりも言葉の扱いが巧みなユーザにとって

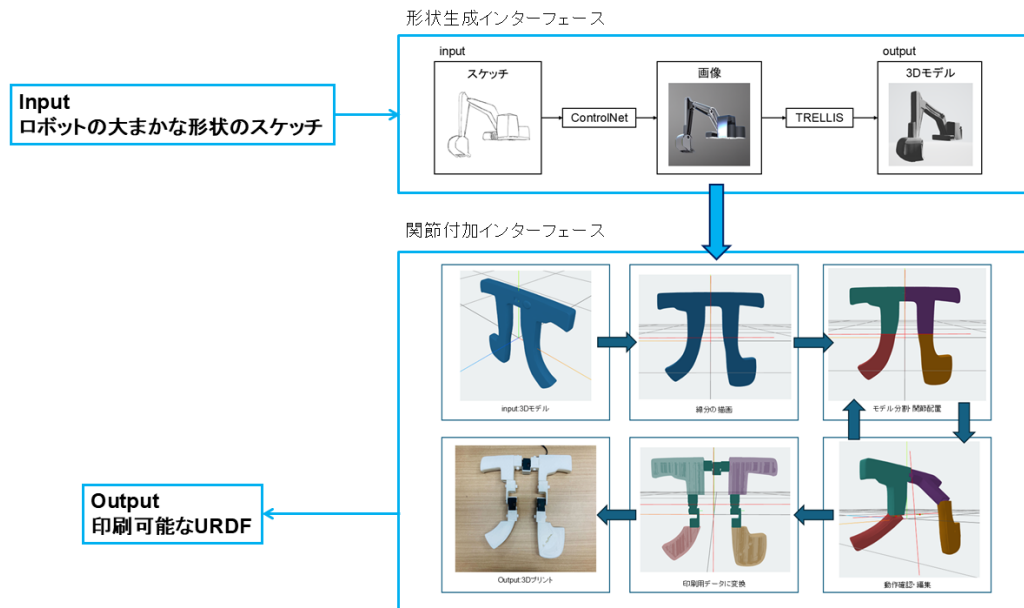


図 5.1: 提案システム Sketch2Robpla

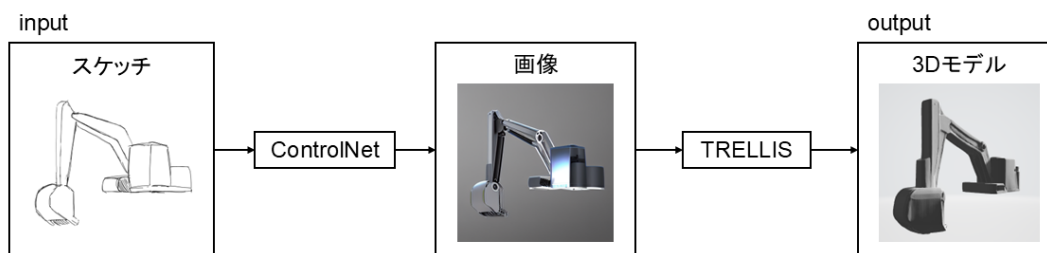


図 5.2: 形状生成インターフェース概要図

非常に有用である。生成ステップを増やすことも画像の品質を上げることにつながるがその分時間がかかるため注意する必要がある。スケッチとプロンプトのバランスを調整したい場合、ガイダンススケールと ControlNet 強度を操作する。プロンプトに書いたことがうまく反映されていない場合はガイダンススケールの値を上げることでプロンプトの影響度が上がる。7-9 程度に設定すると良い。スケッチの線に自信があり、完璧に画像に反映したい場合は ControlNet 強度を上げることでスケッチの影響度が上がる。0.8-1.2 程度に設定すると良い。最後にシードだが、デフォルトはランダムになっているが全く同じ生成設定の場合でもシードを変更することで別の画像が生成される。また、生成候補の画像のシード値を記録しておくことで、生成結果の再現も可能である。画像生成が完了した後は、TRELLIS を用いて 3D モデルを生成する。TRELLIS の場合もシードを変更することで、同一画像から複数の 3D モデルが生成可能であり、シードを控えておくことで再現が可能であるシステムを使って生成した例を図 5.5, 図 5.6 に示す。

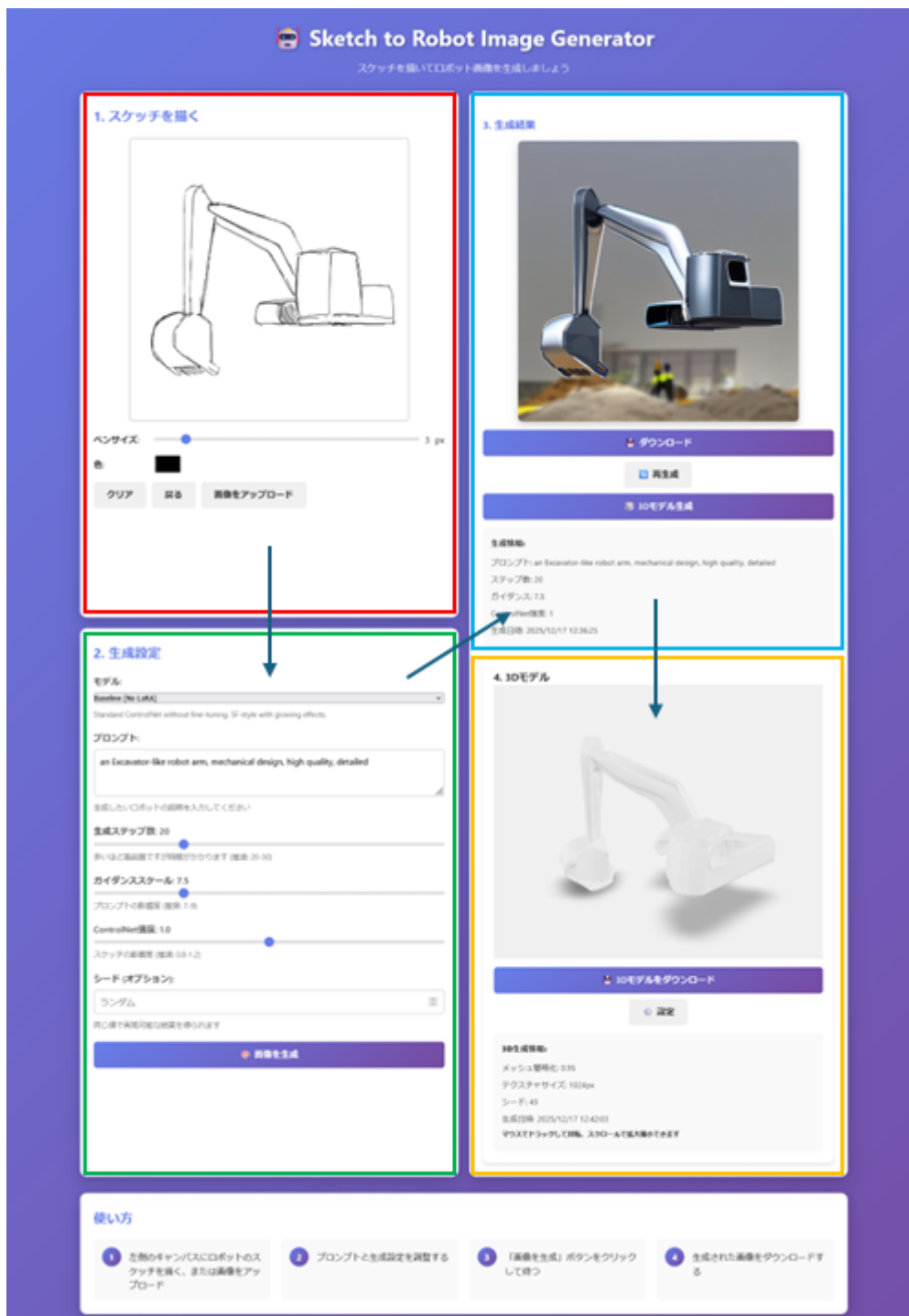


図 5.3: 形状生成インターフェース

2. 生成設定

モデル:

自動選択 (Best Quality) ▾

利用可能な最良モデルを自動選択します

プロンプト:

a long leg robot, mechanical design, high quality, detailed

生成したいロボットの説明を入力してください

生成ステップ数: 20

多いほど高品質ですが時間がかかります (推奨: 20-30)

ガイドンスケール: 7.5

プロンプトの影響度 (推奨: 7-9)

ControlNet強度: 1.0

スケッチの影響度 (推奨: 0.8-1.2)

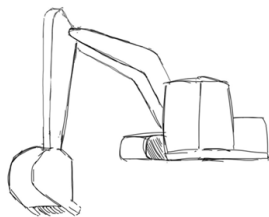
シード (オプション):

42 ▾

同じ値で再現可能な結果を得られます

🎨 画像を生成

図 5.4: 生成設定



(a) 入力スケッチ

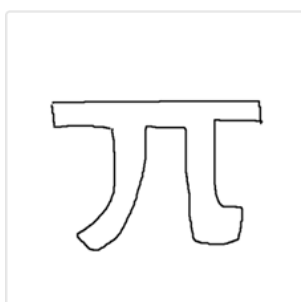


(b) 生成画像

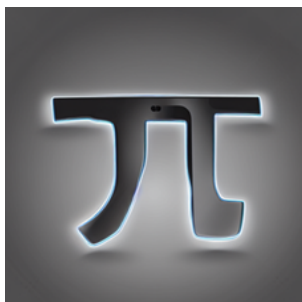


(c) 生成 3D モデル

図 5.5: 生成例 1



(a) 入力スケッチ



(b) 生成画像



(c) 生成 3D モデル

図 5.6: 生成例 2

5.1.2 関節付加インターフェース

ロボットの関節定義を行うインターフェースを図5.8に示す。このインターフェースでは、3Dモデルを読み込み、ユーザが3Dモデルビューに線分を書き込むことでモデルを分割し、断面の中心に関節を配置する。配置した関節の運動学的な動作を確認する。

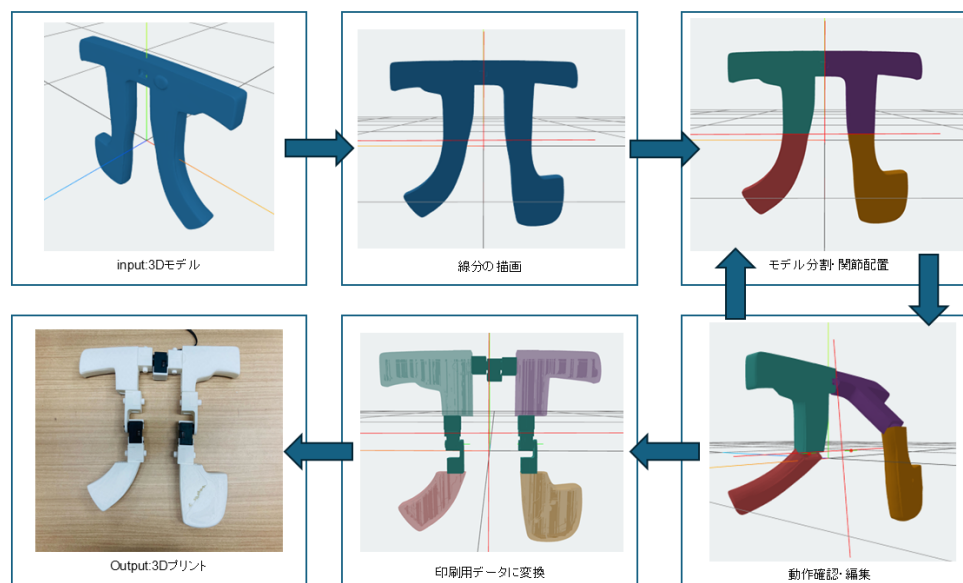


図 5.7: 関節付加インターフェース概要図

3D モデルの読込

まず初めに、UI左上[参照]を押下してファイルダイアログを開き、3Dモデルを読み込む。形状生成インターフェースで生成したモデル以外のモデルを読み込むこともできる。ファイル形式は現在STL,OBJ,GLBに対応している。また、生成された3Dモデルは空洞や破片を含んでいる場合がある。3Dモデル内部の空洞は3Dプリンタで印刷する際に影響があるため、ここでボクセル化して空洞を埋める。

線分スケッチによる3Dモデル分割および関節配置

ロボットの構造入力とは3Dモデル上に線分を引くことで行う。この方法は従来手法の数値入力やSketch2Jointsでのキャンバス上への配置と異なり、ロボットをパーツ分けするように直感的に関節を配置することができる。[Sketch Mode]を押下すると3Dモデルビューのカメラ位置が固定され、ドラッグ操作によって3Dモ

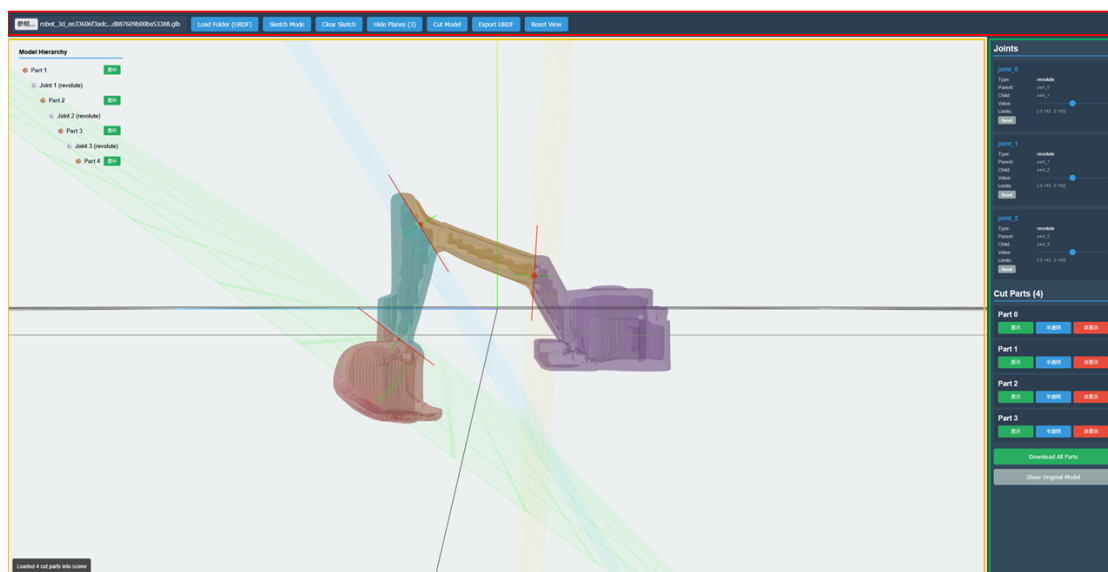


図 5.8: 関節付加インターフェース

デルビュー上に線分を描画することが可能になる．そして [Cut Model] を押下するとユーザの描画した線分の位置で 3D モデルが分割される．ユーザの線分の両端 2 点とカメラ視点の計 3 点から面を定義して分割する．カメラ視点を参照することで，分割結果がユーザの視覚と一致し，ユーザが直感的にモデルの分割を行うことができる．関節定義の従来手法では，関節位置，回転軸を数値で入力する必要があるが，Sketch2Robpla ではモデル分割時にシステムが自動で関節位置，関節回転軸を計算する．分割時，システムは断面の中心を関節位置として関節を配置する．回転軸に関してはユーザの意図が影響する部分であるため，断面中心を通過する最小断面積を持つ断面と切断面の角度を比較し，45 度より大きい場合は切断面に垂直なベクトルを回転軸とし，45 度より小さい場合は切断面に水平なベクトルを回転軸とする．どちらの回転軸を採用するか，また切断面に水平なベクトルの平面上の向きの設定は画面右側の図 5.9 に示すフレームにて編集が可能である．同図下側の [詳細編集] を押下すると，図 5.10 に示すダイアログが表示され，既存手法である数値入力による関節定義が可能である．

関節用メッシュをモデルビューに配置する

分割されたロボットモデルをロボットとして実世界で稼働させるためには，関節として機能する構造が必要である．Sketch2Robpla では関節の機能を持ったメッシュを予め用意し，配置した関節の位置と回転軸を合わせて表示する．このとき，各関節メッシュは回転位置と原点を合わせるように定義しているため，回転軸のみを合わせるためにメッシュの表示位置を幾何変換する．また，配置するメッシュ



図 5.9: 関節情報フレーム

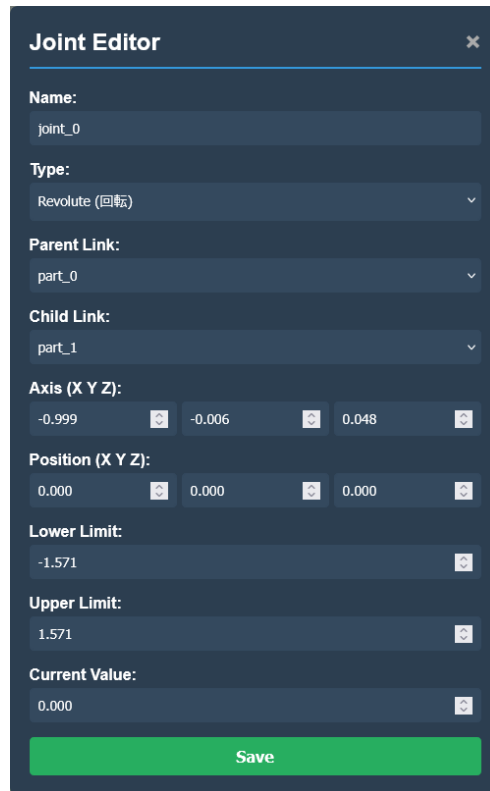


図 5.10: 詳細編集ダイアログ

は図 5.9 に示される関節情報フレームから図 5.11 に示すプルダウンリストを開いて選択する。各関節のメッシュを以下に示す。

- 回転軸の種類として水平を選択している場合の関節の全体を表すメッシュ (full_horizontal_joint.stl)
- 回転軸の種類として垂直を選択している場合の関節の全体を表すメッシュ (full_vertical_joint.stl)
- 回転軸の種類として水平を選択している場合の親側に結合するメッシュ (horizontal_miyajima_parent.stl)
- 回転軸の種類として水平を選択している場合の子側に結合するメッシュ (horizontal_miyajima_child.stl)
- 回転軸の種類として垂直を選択している場合の親側に結合するメッシュ (vertical_miyajima_parent.stl)
- 回転軸の種類として垂直を選択している場合の子側に結合するメッシュ (vertical_miyajima_child.stl)



図 5.11: 関節メッシュプルダウンリスト

また，関節メッシュの回転をロボットモデルが妨げないように，3D モデルの干渉部分を除去する．除去前を図 5.18(a)，除去後を図 5.18(b) に示す．除去範囲は切断面と関節メッシュのバウンディングボックスを用いて決定する．除去範囲は切断面を底面，関節メッシュのバウンディングボックスの切断面に垂直な方向の長さを高さとした直方体として定義し，干渉除去の範囲とする．最後に，関節メッシュとロボットの各パーツを結合して，印刷用の 3D モデルが完成する．

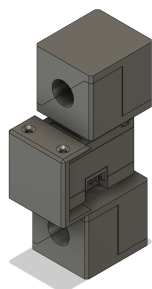


图 5.12: full_Horizontal_Joint.stl

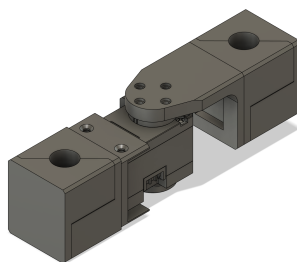


图 5.13: full_vertical_joint.stl

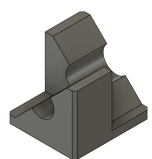


图 5.14: horizontal_miyajima_parent.stl

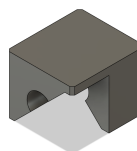


图 5.15: horizontal_miyajima_child.stl

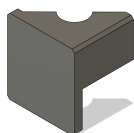


图 5.16: vertical_miyajima_parent.stl

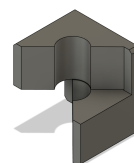
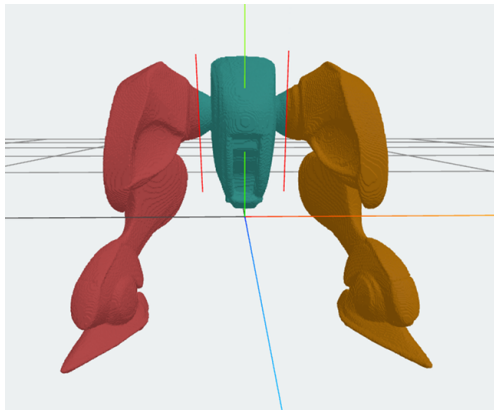
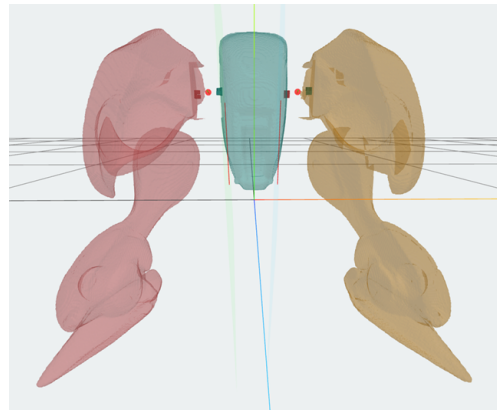


图 5.17: vertical_miyajima_child.stl



(a) 干涉除去前



(b) 干涉除去後

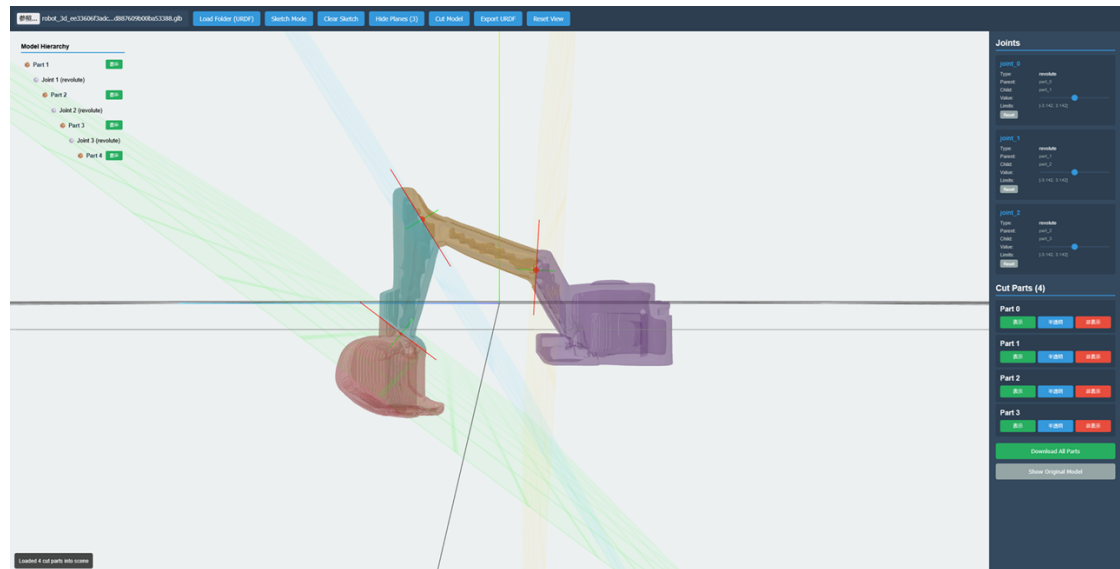
图 5.18: 干涉除去

動作確認・編集

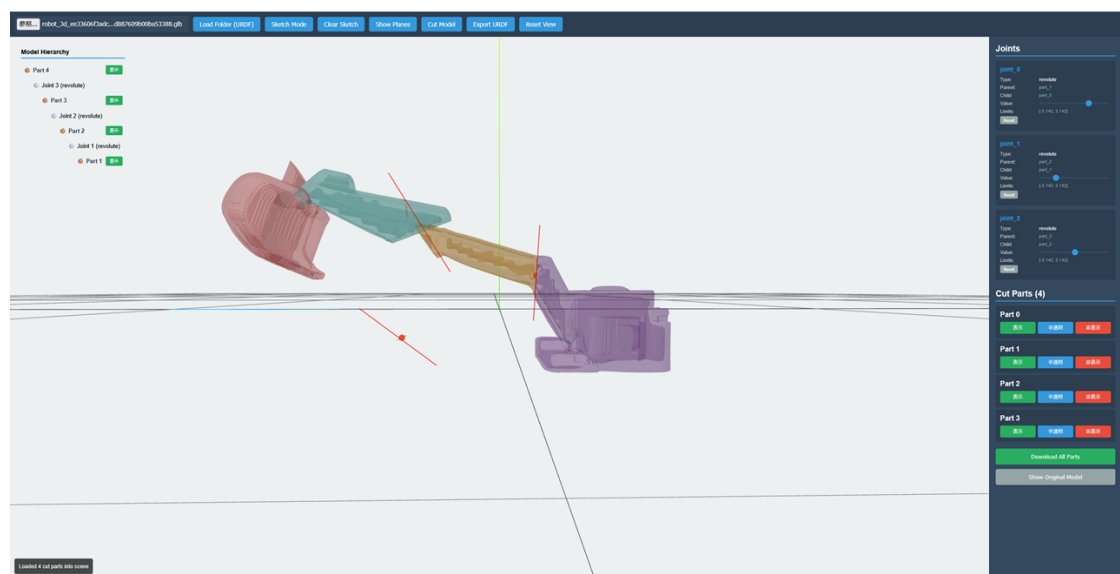
定義した関節は図 5.9 に示す関節情報フレーム内にある Value スライダーを操作することで、変位を変更し、3D モデルビューには変位に応じてロボットの姿勢が変化する (図 5.19(b)). これによりユーザは運動学計算によるフィードバックを受け取り、現在設計しているロボットの動きと頭の中のロボットの動きを比較することができる。比較した結果、イメージと異なる場合は回転軸の編集を行うことができる。図 5.9 の回転軸設定/Axis の右の平行と書いてあるボタンは、回転軸が切断面に平行か垂直かを切り替えることができる。その下の回転と書いてあるスライダーは、タイプが平行の場合に表示され、操作することで回転軸が切断面上を回転する。また、関節位置と同様に図 5.10 に示す詳細編集ダイアログから回転軸を数値入力することもできる。

出力

完成したロボットは URDF ファイルと各パーツのメッシュを出力できる。各パーツのメッシュは、関節メッシュを結合した状態で出力されるため、3D スライサソフトで読み込んで G-code を作成することで 3D プリンタによる製作が可能である。



(a) 関節回転前



(b) 関節回転後

図 5.19: 運動学フィードバック

5.2 結果

5.2.1 ユーザ実験

ユーザ実験として、本学の20代の学生9人を被験者として集めた。被験者達は15分程度の説明を受けた後、Sketch2Robplaを用いてロボットの形状と構造を設計し、アンケートに答えた。ユーザの設計を図5.20 5.25に示す。アンケートの内容はSUS、NASA-TLX、そして以下に示す本研究の課題に関する内容である。

- ロボットを設計した経験について、ある場合は詳細も含む
- 3DCADを操作した経験、ある場合はどの程度か
- 想像通りの形状を生成できたか
- 想像通りの構造を設計できたか

SUSの結果を図5.26に示す。「このシステムを頻繁に使用したい」「このシステムは必要以上に複雑だと思う」「このシステムはスムーズに機能し、連携がとれていると思う」「ほとんどの人がこのシステムをすぐに習得できると思う」「このシステムを使っていると自信が持てる」これらの項目に関しては、良い結果が得られた。特に結果が悪かった項目は、「このシステムにはテクニカルサポートが必要」である。ここから、UI上に操作方法の示唆が足りていないことが考えられる。また、主な操作である線分による分割時にユーザの意図と異なる切断が発生した点や、切断後の接続関係が正常でない場合があった点、その際に接続関係の編集方法が直感的でなく、専門性も高かったことから、テクニカルサポートが必要だったことは間違いない。「このシステムには不規則な点が多いと思う」「このシステムには時間がかかると思う」「このシステムを使う前に学ぶべきことはたくさんあると思う」これらの項目は被験者によって評価が分かれた。また、「このシステムはシンプルで使いやすい」に関してもわずかに悪い評価を受けた。これらの結果は、操作方法の明確な指示、必要な情報の提示ができておらず、また、ロボット設計を作る上で必要になる関節位置、回転軸、接続関係の決定に関して、予期しない動作が多く見られ、それらの編集方法が古典的手法と変わらない数値入力であったことが原因であると考えられる。

次に、NASA-TLXの結果を図5.27,5.28に示す。作業成績のスコアは既に反転してあるため、スコアが高い方が満足いく成績を残せなかったことを表している。総合スコアの平均値は48.9であり、作業負荷は中程度であった。項目別に見ると、タイムプレッシャーは特に時間制限を設けなかったことからこの結果になったことが考えられる。身体的要求に関してもシステムにおいて身体に負荷のかかる作業は存在しないため低い結果になったと考えられる。知的・知覚的要求が少し高めの値であるが、被験者のほとんどが3DCAD、ロボット設計の未経験者であること

が原因の一つであると考えられる。システムの持つ問題は作業成績、努力、フラストレーションの項目に表れている。その中で、作業成績、フラストレーションの項目については回答の幅が広い。この振れ幅は実験中に不具合がどれだけ発生したかが影響している可能性がある。特に、複雑な 3D モデルを生成した被験者は不具合の発生により満足な関節構造を設計できないことが多かった。最後に努力の項目だが、これが一番高い負荷を示していた。そして、努力スコアが高い被験者は、作業成績、知的・知覚的欲求のスコアも高い傾向にあるため、ユーザがタスクに関する意欲を失っていることや、複雑な計算、判断を強いられている可能性がある。

5.2.2 製作

システムで設計したロボット (図 5.29(a)) を 3D プリンタ (FlashForge Adventurer 5M) で印刷した結果が図 5.29(b) である。製作にかかった時間はロボット本体がパーツ 4 つで 8 時間 14 分、関節が水平回転関節 1 つと垂直回転関節 2 つで 1 時間 23 分、計 9 時間 37 分であった。

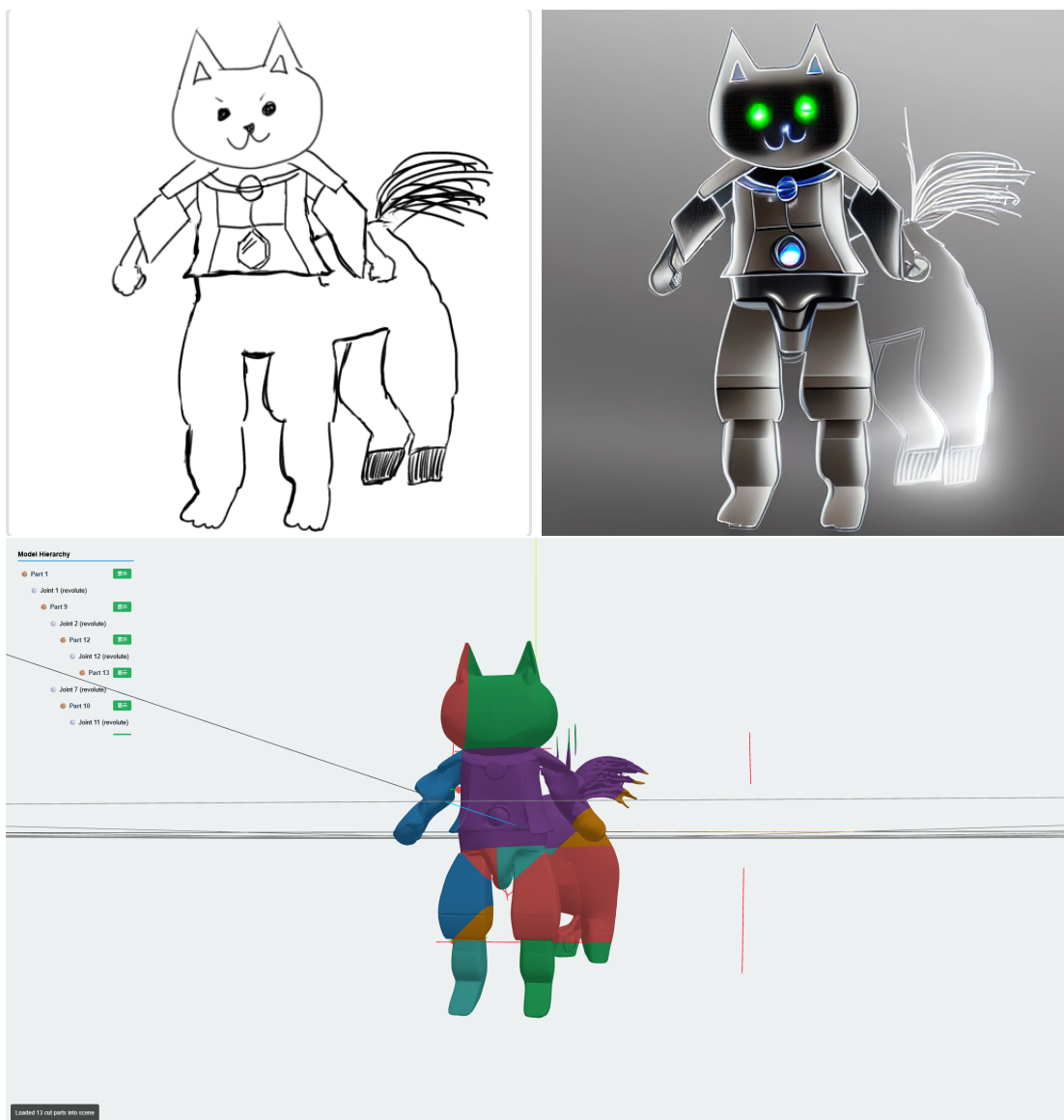


図 5.20: ユーザ A による製作結果

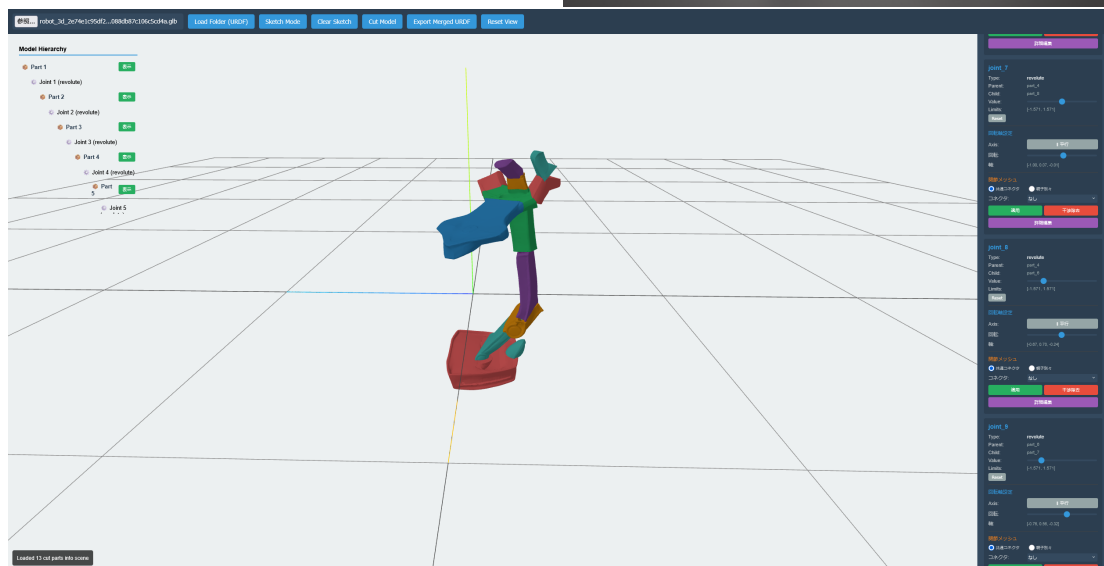
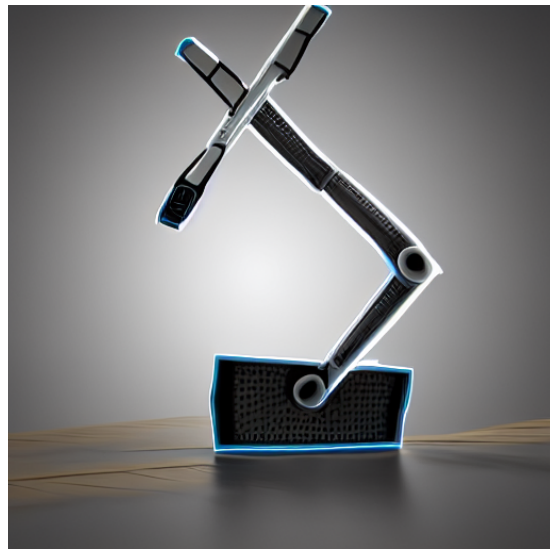
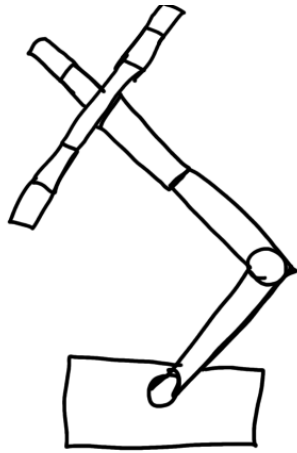


図 5.21: ユーザ B による製作結果

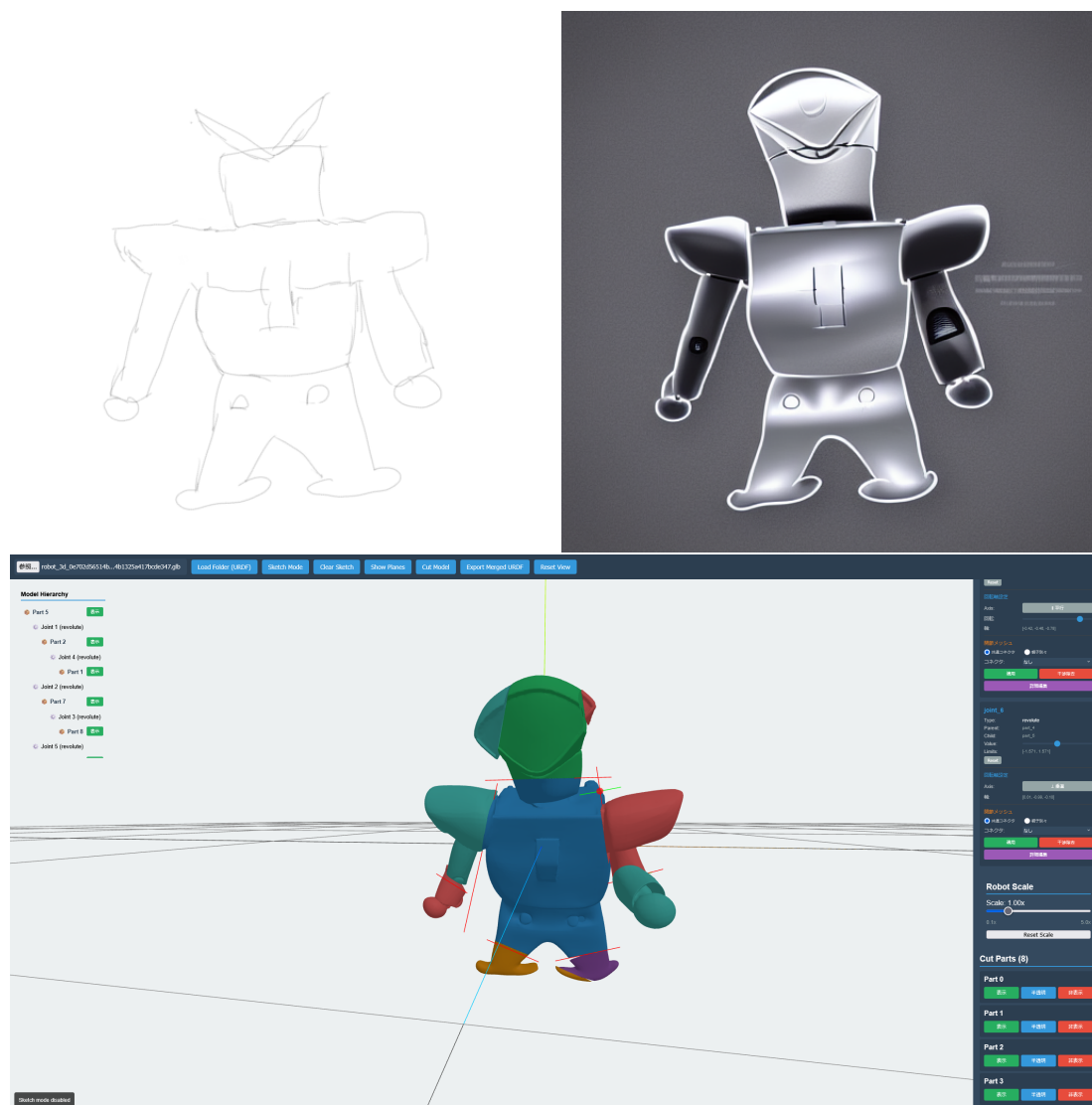


図 5.22: ユーザ C による製作結果

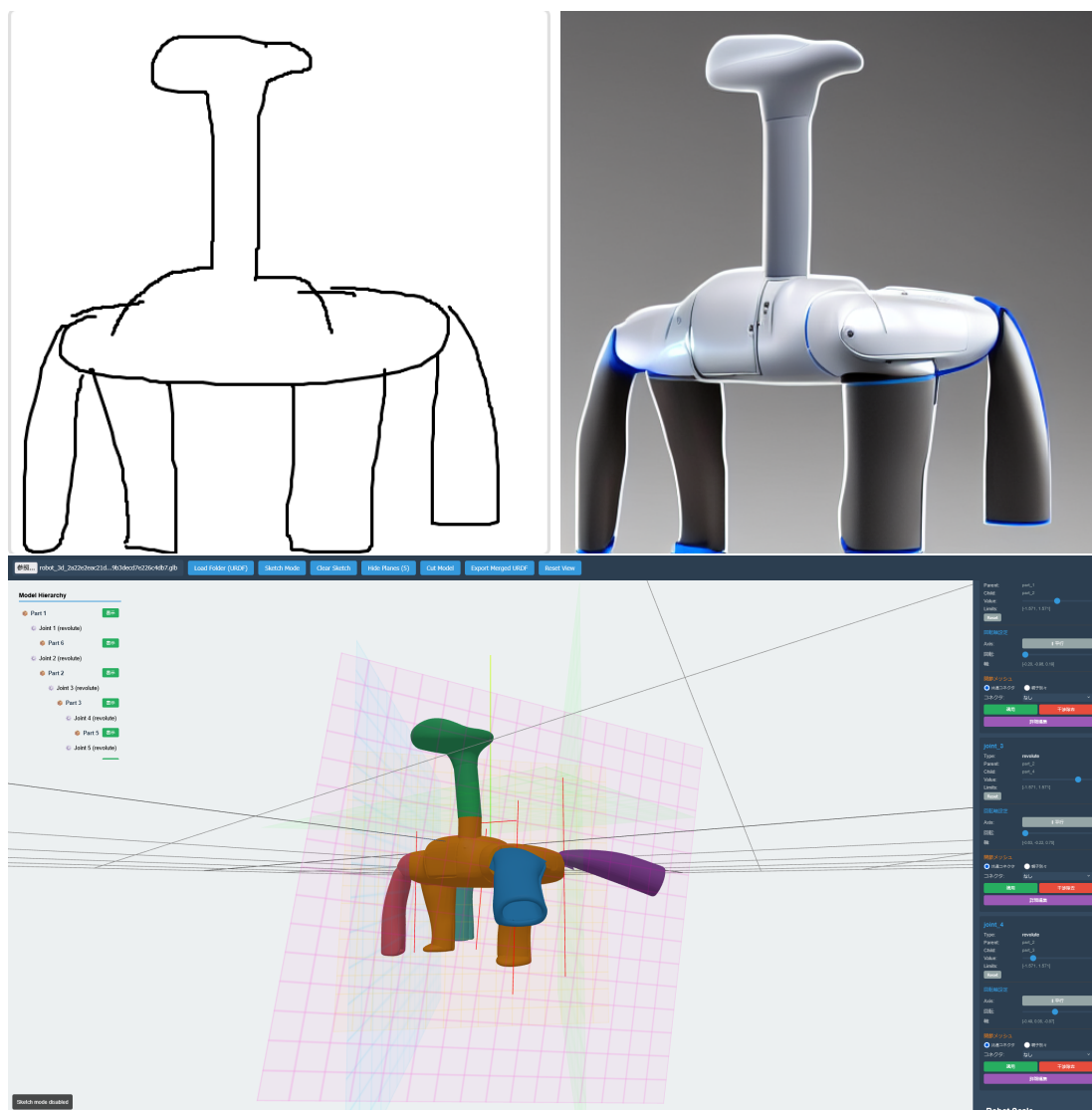


図 5.23: ユーザ D による製作結果

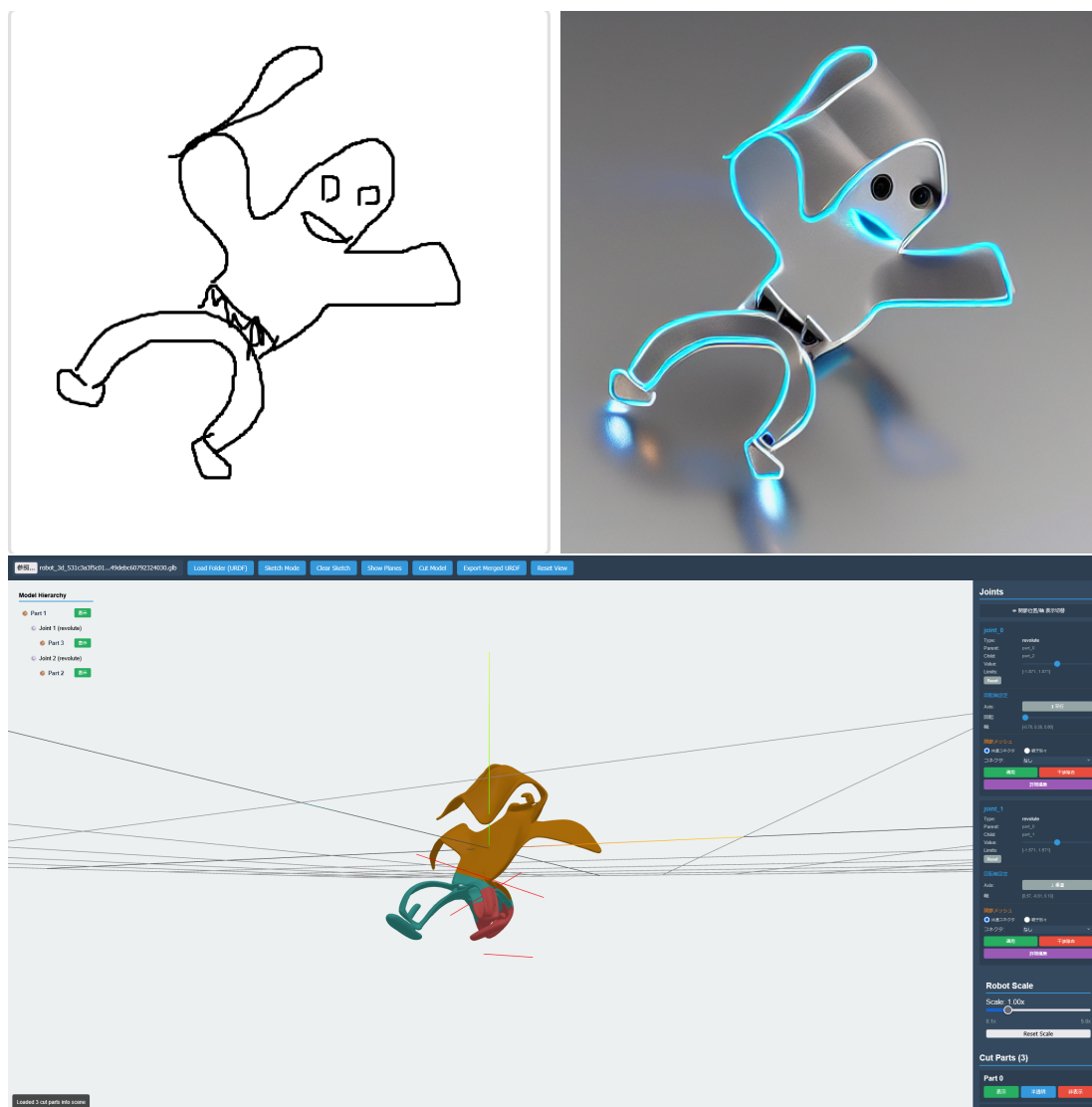


図 5.24: ユーザ E による製作結果

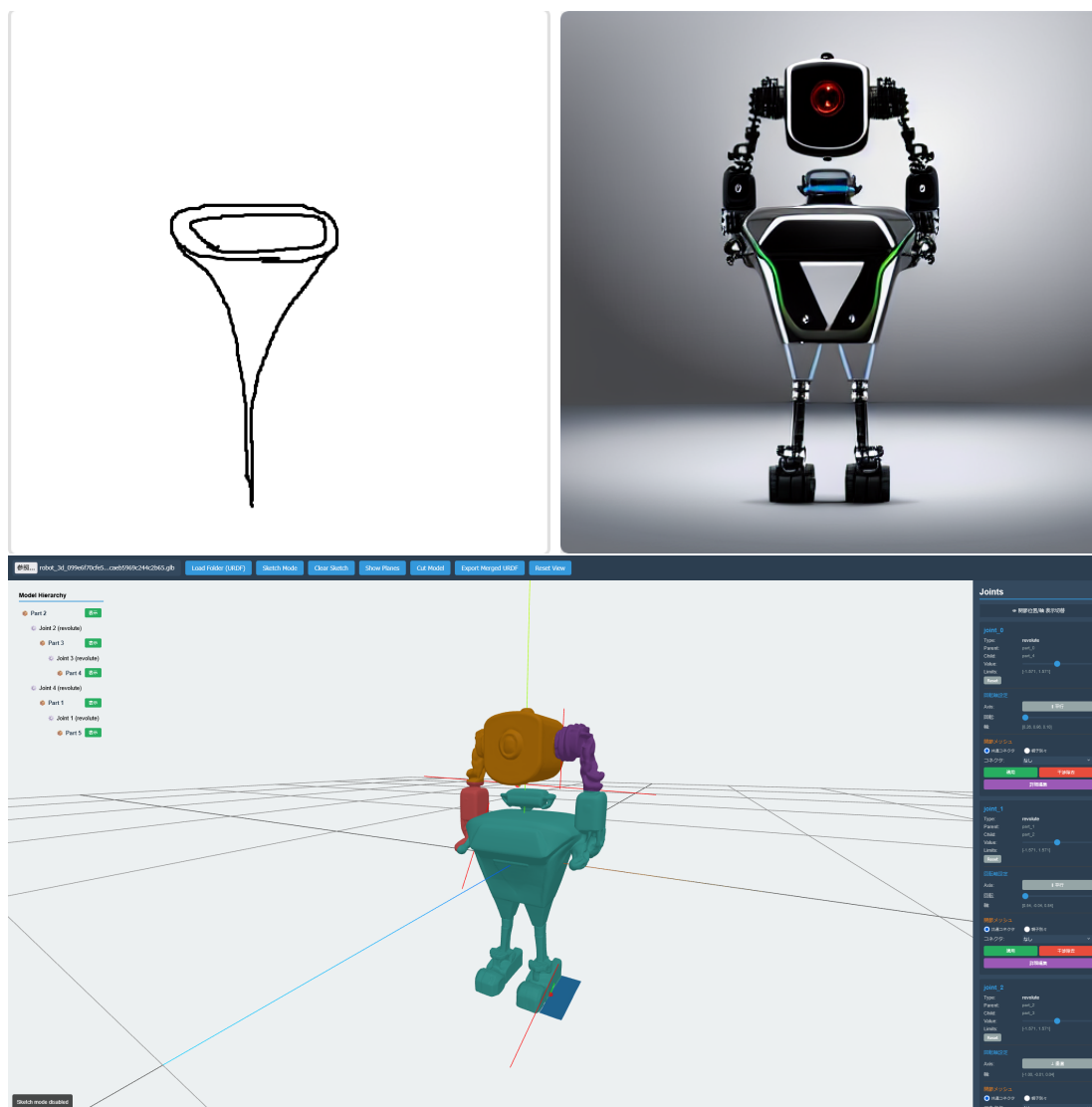


図 5.25: ユーザ F による製作結果

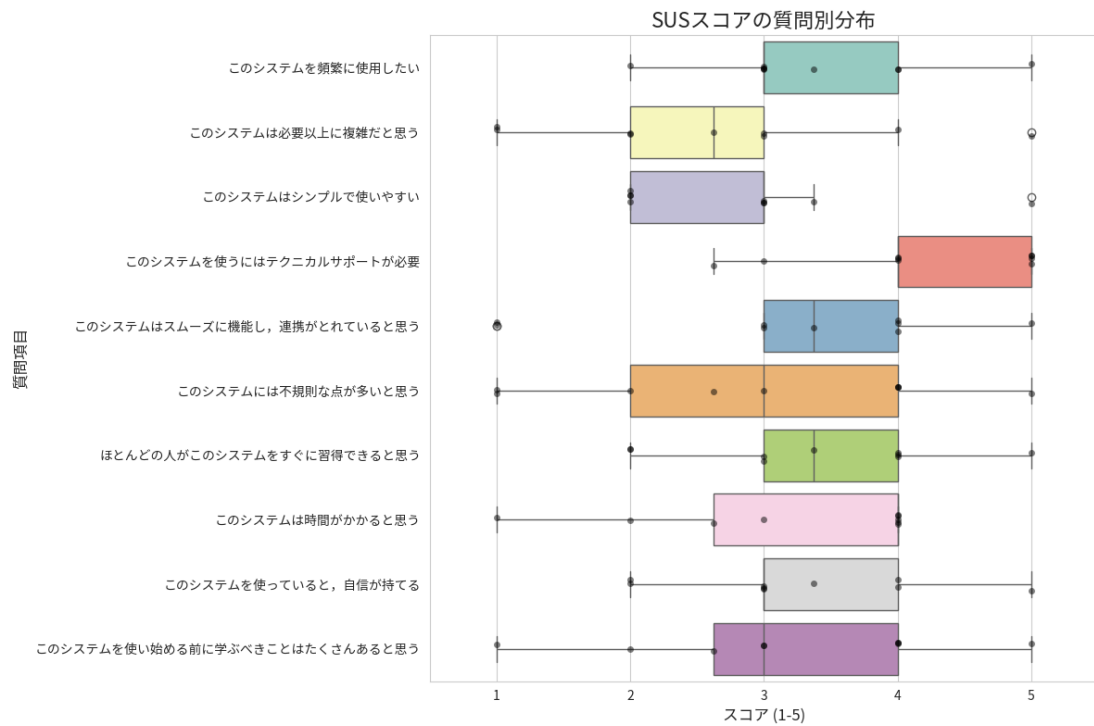


図 5.26: System Usability Scale の結果

NASA-TLX平均

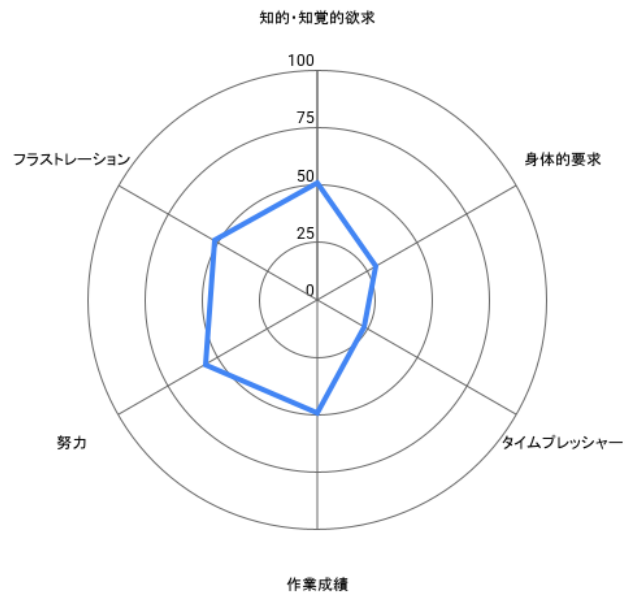


図 5.27: NASA-TLX の結果 (平均)

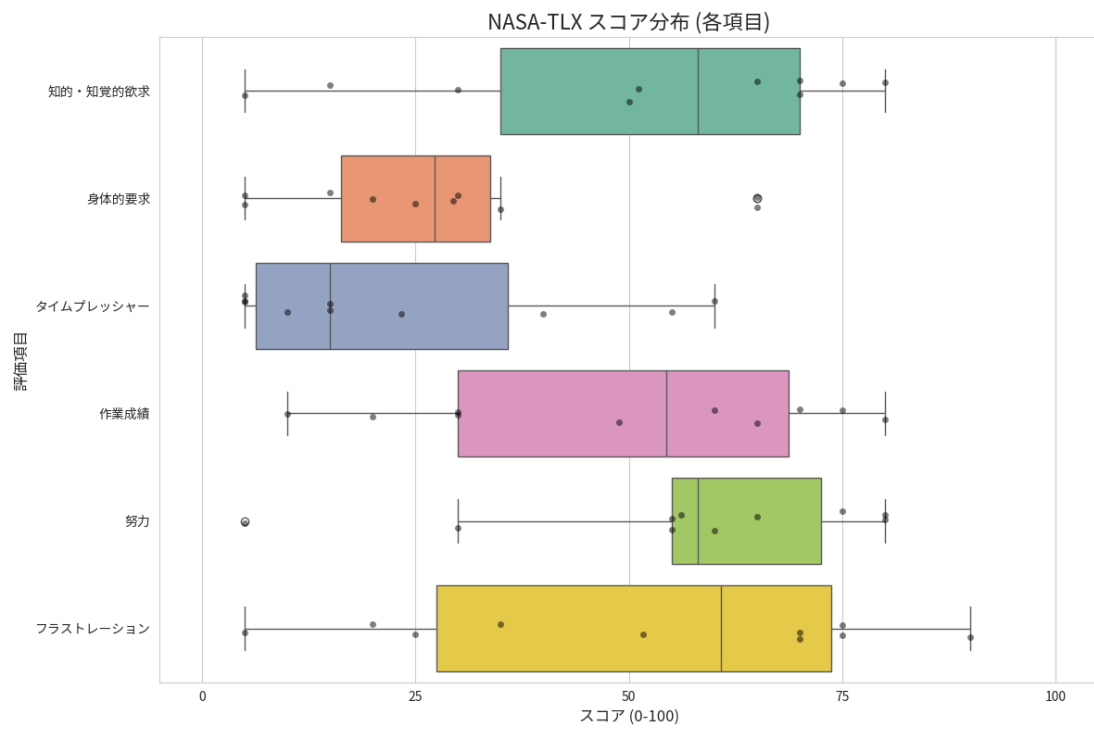
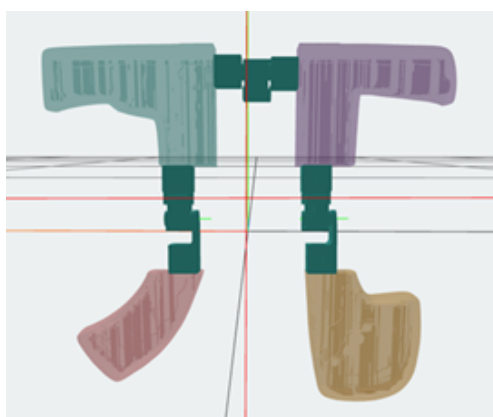


図 5.28: NASA-TLX の結果 (項目別)



(a) 印刷データ



(b) 印刷結果

図 5.29: 製作結果

第6章 おわりに

本研究ではロボット設計の民主化を目指し、スケッチベースインターフェースによるロボット設計システムを開発した。はじめに関節を記号化してキャンバス上に配置することで関節構造を設計し、関節構造から自動的にリンクを作成、3Dプリンタで印刷するシステム、Sketch2Jointsを開発した。これにより形状を反映した運動学フィードバックや直感的な関節構造入力、より形状自由度の高い形状設計手法の必要性が判明した。この問題の解決を目的とし、生成AIを用いて高い形状自由度を実現し、線分の描画によるモデル分割及び関節配置によって関節構造入力をより直感的にし、運動学フィードバックに生成したモデルを反映したシステム、Sketch2Robplaを開発した。

6.1 課題

Sketch2Robpla を用いて行ったユーザ実験により、現状の Sketch2Robpla には、ユーザビリティの欠如や動作の不安定性といった問題が見つかった。これらを改善するために、ユーザへの具体的な操作方法の提示、関節とリンクの接続関係入力、関節位置の編集操作を直感的な操作で実現することにより、これらの問題を解決する必要がある。また、生成した画像、3D モデルの一部編集機能を実装、3DMMLM による関節推定の実装によってシステムをさらなる改善が見込める。

6.2 展望

本研究は、ロボット設計の経験が無いユーザが直感的にロボット設計を行うことを目指した。このロボット設計の民主化が達成されれば、安価なロボット製作が可能になり、一般家庭へのロボットの普及だけではなく、特定の環境、タスクに応じたロボット設計によって、災害現場で作業するロボットや伝統産業の支援ロボットの開発に貢献できる。また、データ収集にも貢献できると考える。近年では、データ駆動型のロボット制御手法が盛んに研究されており、特に身体に依らないロボット制御ポリシーを獲得できるクロスエンボディメント学習はロボット制御の民主化に必要な技術である。Sketch2Robpla はクロスエンボディメント学習に用いるためのロボットデータを取得することができる。既に RoboGrammar のグラフ文法で設計したロボットによる学習が行われており、クロスエンボディメ

ント学習におけるスケーリング則が実証されている [31]. この研究では精巧ではないロボットデータの学習によって制御ポリシーの堅牢性が向上することも示されており, 初心者が設計したロボット, あるいはそのロボットを基に生成したロボットを学習データに加えることによって制御ポリシーの性能向上につながると考えられる.

第7章 謝辞

本研究を進めるにあたって、適切な助言とご指導をしてくださった北陸先端科学技術大学院大学の謝浩然准教授に深く感謝します。また、論文の執筆方法、添削、プレゼンテーション方法そして研究目的や研究の意義についての考え方などをご教授いただき心よりお礼申し上げます。

また、研究室の皆様におかれましては、研究における私の初歩的な疑問への回答、私の研究に対する適切なご指摘、助言を頂き本当にありがとうございます。

そして、ユーザ実験に協力していただいた皆様に対しても感謝申し上げます。

参考文献

- [1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [2] Inc. Autodesk. Autodesk fusion, 2024.
- [3] Dassault Systèmes. Solidworks, 1995.
- [4] Adriana Schulz, Cynthia Sung, Andrew Spielberg, Wei Zhao, Robin Cheng, Eitan Grinspun, Daniela Rus, and Wojciech Matusik. Interactive robogami: An end-to-end system for design of robots with ground locomotion. *The International Journal of Robotics Research*, 36(10):1131–1147, 2017.
- [5] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 1321–1326. IEEE, 2013.
- [6] Zhengyu Huang, Haoran Xie, Tsukasa Fukusato, and Kazunori Miyata. Anifacedrawing: Anime portrait exploration during your sketching. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery.
- [7] Xusheng Du, Yi He, Xi Yang, Chia-Ming Chang, and Haoran Xie. Sketch-based 3d shape modeling from sparse point clouds. In *International Workshop on Advanced Imaging Technology (IWAIT) 2022*, volume 12177, pages 714–719. SPIE, 2022.
- [8] Haoran Xie, Keisuke Arihara, Syuhei Sato, and Kazunori Miyata. Dualsmoke: Sketch-based smoke illustration design with two-stage generative model. *Computational Visual Media*, pages 1–15, 2024.
- [9] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

- [10] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W Sumner, Wojciech Matusik, and Bernd Bickel. Computational design of mechanical characters. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.
- [11] Marek Dvorožňák, Daniel Šykora, Cassidy Curtis, Brian Curless, Olga Sorkine-Hornung, and David Salesin. Monster mash: a single-view approach to casual 3d modeling and animation. *ACM Transactions on Graphics (ToG)*, 39(6):1–12, 2020.
- [12] Kosei Tanada, Yuka Iwanaga, Masayoshi Tsuchinaga, Yuji Nakamura, Takemitsu Mori, Remi Sakai, and Takashi Yamamoto. Sketch-moma: Teleoperation for mobile manipulator via interpretation of hand-drawn sketches. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4753–4760. IEEE, 2025.
- [13] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.
- [14] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.
- [15] Yunhan Yang, Yufan Zhou, Yuan-Chen Guo, Zi-Xin Zou, Yukun Huang, Ying-Tian Liu, Hao Xu, Ding Liang, Yan-Pei Cao, and Xihui Liu. Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025.
- [16] Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. Llama-mesh: Unifying 3d mesh generation with language models. *arXiv preprint arXiv:2411.09595*, 2024.
- [17] Shuangkang Fang, I Shen, Yufeng Wang, Yi-Hsuan Tsai, Yi Yang, Shuchang Zhou, Wenrui Ding, Takeo Igarashi, Ming-Hsuan Yang, et al. Meshllm: Empowering large language models to progressively understand and generate 3d mesh. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14061–14072, 2025.
- [18] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language

- models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.
- [19] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pages 214–238. Springer, 2024.
 - [20] Junliang Ye, Zhengyi Wang, Ruowen Zhao, Shenghao Xie, and Jun Zhu. Shapellm-omni: A native multimodal llm for 3d generation and understanding. *arXiv preprint arXiv:2506.01853*, 2025.
 - [21] Vittorio Megaro, Bernhard Thomaszewski, Maurizio Nitti, Otmar Hilliges, Markus Gross, and Stelian Coros. Interactive design of 3d-printable robotic creatures. *ACM Transactions on Graphics (TOG)*, 34(6):1–9, 2015.
 - [22] Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Robogrammar: graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.
 - [23] Kevin Qiu, Władysław Pałucki, Krzysztof Ciebiera, Paweł Fijałkowski, Marek Cygan, and Łukasz Kuciński. Robomorph: Evolving robot morphology using large language models. *arXiv preprint arXiv:2407.08626*, 2024.
 - [24] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023.
 - [25] Genesis Authors. Genesis: A generative and universal physics engine for robotics and beyond, December 2024.
 - [26] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.
 - [27] Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Automated dense reward function generation for reinforcement learning. In *International Conference on Learning Representations (ICLR), 2024 (07/05/2024-11/05/2024, Vienna, Austria)*, 2024.

- [28] Daniella Tola and Peter Corke. Understanding urdf: A dataset and analysis. *IEEE Robotics and Automation Letters*, 9(5):4479–4486, 2024.
- [29] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.
- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [31] Bo Ai, Liu Dai, Nico Bohlinger, Dichen Li, Tongzhou Mu, Zhanxin Wu, K Fay, Henrik I Christensen, Jan Peters, and Hao Su. Towards embodiment scaling laws in robot locomotion. *Conference on Robot Learning (CoRL)*, 2025.