

# Parametric Robot Control: Integrated CAD/CAM for Architectural Design

## 備考

## 著者

Johannes Braumann, Sigrid Brell-Cokcan

## 掲載

"Parametric Robot Control: Integrated CAD/CAM for Architectural Design," Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), pp. 242-251, 2011.

## メモ

パラメトリックデザインは、3Dモデリングソフト上で、設計者の意図する条件=パラメーター（数値変数）を設定して数値を変えていくことで、膨大なデザインのバリエーションを生み出すことができる3次元の設計手法。

## Abst

建築の分野でもロボットの人気が高まっています。Snøhetta は最近、自社製の産業用ロボットを購入し、ロボット技術を採用した最初の建築事務所の一つとなりました。より多くの建築家がロボットによる製作に触れるようになると、相互運用性、建築設計ツールへの統合、そして一般的なアクセシビリティの必要性が高まります。建築家は、産業用ロボットがレンガを積み上げたり、溶接したり、フライス加工したりするための運動機械以上のものであり、非常に多機能で、非常に多様な作業に使用できることを知っています。しかし、業界標準のソフトウェアでは、CAAD (Computer Aided Architectural Design) システムから直接ロボットを制御するための簡単なソリューションは提供されていません。この論文では、産業用ロボットをプログラミングするための既存の方法、発表された建

築の成果 (Gramazio and Kohler 2008)、およびCAM (Computer Aided Manufacturing) 機能を CAAD に統合することで、パラメトリックデザインを直感的に制御し、ロボットによるカスタマイズされた大量生産を可能にする新しいユーザーインターフェースの設計について説明します。

# 1. はじめに

ここ数十年の CAAD ソフトウェアの急速な発展により、建築物の形状をパラメトリックに制御できるようになり (図1)、その結果、デザインバリエーションの自動生成や全体的なデザイン管理が可能になりました (Eastman et al.2008)。パラメトリックデザインを実現するためには、マスカスタマイゼーションと呼ばれる、合理的な低コストで大量に製造できる柔軟なプロセスが必要です (Piller 2004; Ramani et al.) 建築家は、自らのデザインの建設と製作のプロセス、つまり機械的処理をコントロールしたいと考えている (Gramazio & Kohler, Menges, Scheurer in Kolarevic 2008; Iwamoto 2009)。パラメトリックな自由形状のデザインが具体化されると同時に、一つの問題が発生します。それは、建築家がデザインプロセス全体の最終段階であるファブリケーションをコントロールするための適切なソフトウェアを持っていないことです。**主要な CAAD 開発者は、これまでのところ、さまざまなタイプの機械を CAD システムに直接リンクさせるための適切なソリューションを提供していません。自由形状の建築に携わる建築業界は、この欠点を「自社製」のカスタムメイドのソフトウェアツールを開発することで克服してきました。**そのため、建築におけるハイエンドの形状と製作のコンサルティングは、コンピュータ科学者や数学者が運営する新しい専門的なコアビジネスとして急速に普及しています (Evolute (Eigensatz et al. 2010)や design-to-production (Scheurer 2010) などの企業を参照)。私たちは以前の研究で、パラメトリックオブジェクトのデザインだけでなく、その製造関連の特性を評価するために、製造に内在するデザインとリアルタイムのフィードバックの利点を示しました (Brell-Cokcan and Braumann 2010)。本論文では、産業用ロボットの新しいインターフェースとして、グラフィカルアルゴリズムエディタの使用を提案する。その結果、ユーザーがカスタマイズした CAAD ソフトウェアでは、機械の制約条件が追加の設計パラメータとして含まれ、ロボットの制御データファイルが直接生成され、マスカスタマイゼーションに利用される。

# 2. KUKAロボットの既存のプログラミング方法

一般的に、産業用ロボットのプログラミングには2つの特徴的な戦略があります。オンライン/自動 (図2) と オフライン/手動のプログラミング (図3) です (Biggs and MacDonald 2003)。オンライン・プログラミングは、主に荷役や溶接などの産業用アプリケーションに使用され、KCP (KUKA Control Panel) や仮想ロボット環境の KUKA SimPro を使ってポイントを個別に「教示」します。**デジタル**

CAD/CAM ワークフローが重要な建築分野では、オンラインプログラミングはほとんど使用されていません。

## 2.1. ロボットオフラインプログラミング

ロボットのオフラインプログラミングは、ロボットセルの外にある外部コンピュータで行われ、従来のCNC (Computer Numeric Control) マシンでの作業方法 (CADからCAMソフトウェアにデータを移動し、NC(Numeric Control)コード(G-Codeとしても知られている)としてCNCフライス盤に移動する方法) に似ています。さらに、ロボットのミリングには、キネマティック・ポストプロセッサ/シミュレーター (当社の場合はKUKA SimPro/CAMRob) の使用が必要です。この結果、異なる開発者による少なくとも3つのソフトウェアツールを含むデジタルワークフローとなります。**ワークフロー全体を通して製造特性を評価できるような、"リアルタイム"の設計・製造フィードバックが欠けています。** 先行研究では、ロボットの加工にパラメトリック設計ツールを使用することで、デジタルCAD-CAMロボットのワークフローを効率化できる可能性が示されている (Brell-Cokcan et al.). KUKAロボットのオフラインプログラミングをさらに最適化するために、セクション4では、**CAD ソフトウェア Rhinoceros のパラメトリックデザインプラグインである Grasshopper** 内で、KUKA|prc (パラメトリックロボットコントロール) のカスタムコンポーネントを設計する方法を紹介します。

## 2.2. kuka ロボットの言語コード

KUKA Robot Language (KRL) は、Pascal に似た独自のプログラミング言語で、KUKA ロボットを制御するために使用されます。Gコードとは異なり、工具や機械の動作命令だけでなく、変数の宣言や条件文を扱うこともできる。KRL のドキュメント (KUKA Robotics 2003) では、一連のコマンドについては説明されているものの、以下の適切な制御データファイルの必要な構造についての詳細な情報は提供されていない。KRL (Mühe et al. 2010) を使用しています。そのため、キネマティクスของポストプロセッサである KUKA SimPro/CAMRob によって作成されるKRLファイルの構造をリバースエンジニアリングする必要がありました。基本的に、KUKA用の数値制御ファイルは2つの別々のファイルで構成されています。datファイルにはホームポジションなどの変数が格納され、\*.srcファイルには動作コマンド、ループ、センサクエリなどの様々なコマンドが格納されています。ロボットへの動作命令の与え方には様々な方法があるが、最も一般的なものは次のようなものである (Takase et al 1981):

関節座標のプログラミングは、絶対軸の回転命令で構成されており、定義された回転値に合わせてロボットの6軸それぞれを動かすように指示します。

A1 0、A2 10、A3 90、A4 20、A5 60、A6 25

直交座標プログラミングでは、あらかじめ定義された直交座標系におけるエンドエフェクタの位置と向きを定義します。これには、エンドエフェクタがある位置から次の位置まで軸の回転を最小限に抑

えて移動するポイントツーポイント (PTP) コマンドと、エンドエフェクタがある位置から次の位置まで直線に沿って移動するリニア (LIN) コマンドがあります。

x 10, y 20, z 40, a 45, b 75, c 15

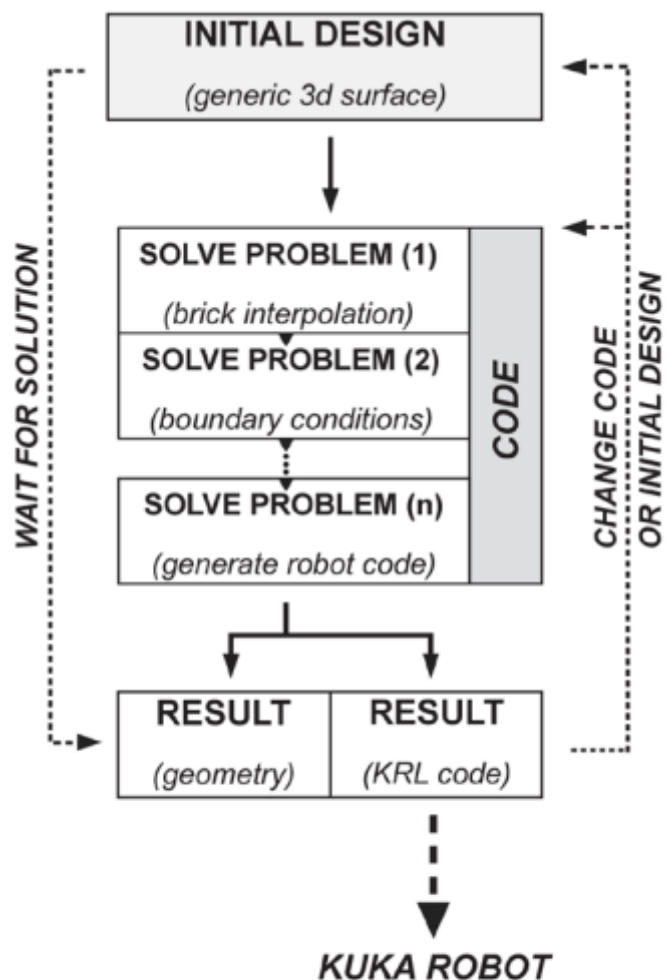
### 3. 建築設計のためのパラメトリック・ロボット・プログラミング

第2章で説明したロボットのデータ処理を理解すると、カスタムスクリプト (オフラインプログラミング) でKRLコードを作成することが可能になり、建築のマスカスタマイゼーションのための製作プロセスを自動化することができるようになります。

ほとんどの建築用途では、ツールの位置を正確に制御する必要があるため、直交座標プログラミングを使用する必要があります。ロボットの運動学は、制御ユニットでKRLコードが実行される際に、エンドエフェクタの位置や向きに応じてリアルタイムに解決されるため、必ずしもパラメトリックモデリング環境でシミュレーションを行う必要はありません。しかし、設計環境でロボットの運動学をシミュレートすることで、設計者は到達可能性の制約や衝突の可能性に対応することができ、利用可能なワークスペースの使用を最適化することができます。

### 4. ロボット・ミリングのためのパラメトリック・コンポーネントの設計

ロボットでの作業を容易にするための主なアイデアは、ロボットのミリングストラテジーが幾何学的なデザインを最初から決定するというボトムアップ型のデザインであり、したがって、分析とシミュレーションのために常にバックグラウンドで利用可能な、デザインの不可欠な部分となります。数回のマウスクリックだけで、CADからCAM、ロボットへの複数のエクスポート/インポートステップを経ることなく、現在のモデルの無限のデザインバリエーションをいつでもロボットで製造することができます (図3)。衝突や工具長の不足は実行時に検出されるため、設計 (=コード) を即座に修正することができます。パラメトリック環境としてGrasshopperが選ばれたのは、モジュール式のオープンな構造とリアルタイムのプレビュー機能を備えているからです。ロボットミリングにおいて特に重要なのは、Grasshopperのコンポーネントシステムが、新たな設計要件や問題に対応するために、ソリューションを区分けして既存のコードを再利用できることです。このようなコンポーネントの機能グループをモジュールと呼んでいます (図5b)。



ig. 5a

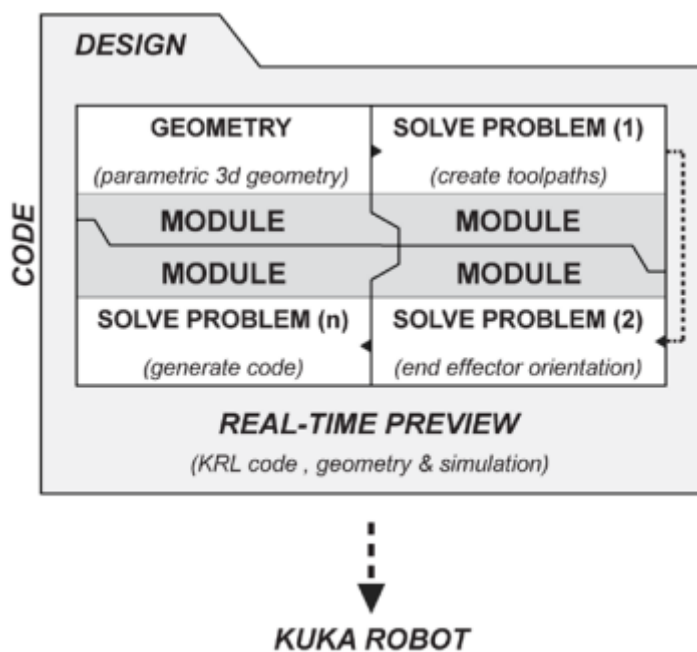


図5a. データフローの比較. 線形プロセスと非線形プロセス: 従来のスクリプティング

図5b. ロボットミリングのための統合パラメトリックアプローチ

例えば、設計者は、複数のジオメトリモジュールを作成し、それらをツールパス作成モジュールに順次差し込むだけで、エンドエフェクタの向きの計算、衝突のチェック、KRLコードの作成を同時に行い、直ちにツールパスを生成することができます。「従来のスクリプティングは、明確に定義された問題を、あらかじめ定義された順序で解決する」ものと考えられますが、これはほとんどのGrasshopperモジュールにも当てはまります。ロボットの制約やツールパス生成などの重要なCAM機能をパラメトリックモデルに統合することで、ロボットミリングのデザインツールを生成し、すべての変更に対応し、いつでもどのモジュールでも流動的で直感的な修正を可能にします。これにより、何百行ものコードや、複数の異なるCAD、CAM、キネマティックソフトウェアパッケージやポストプロセスを使用することなく、ロボットミリングのデザインを素早く評価し、ツールパスをシミュレーションし、加工コードを生成することができます。このように、KRLコンポーネントは、CAADからCAM、ロボットへのフィードバックループを確立します。

## 4.1. kuka krl-component: 一般的なアプローチ

Grasshopper用の KUKA Robot Language コンポーネントを設計するにあたり、以下の5つの要件を満たすことを目指しました。

- コンポーネントは、ネイティブなコンポーネントのようにGrasshopperの環境に統合されなければなりません。
- 市販のCAMソフトのような機能過多ではなく、柔軟でダイナミックなロボット制御に焦点を当てなければなりません。
- その計算は、リアルタイムに近いフィードバックを維持するために1秒以内に完了するようになっており、さらにクイックプレビューモードも用意されています。
- その機能は、デザイナーにとって透過的であり、カスタマイズ可能であり、アクセス可能なユーザーインターフェースに包まれていなければなりません。
- 出力は、人間が読める、バイナリではないKRLコードで構成されていなければなりません。

は、後処理のような追加のステップなしに、ロボットで直接実行することができます。KUKA Robot Language Component for Grasshopper は、Visual Studio の.NETフレームワーク (RhinoCommon SDK & Grasshopper SDK) を使用して記述され、デフォルトの Grasshopper コンポーネントとシームレスに統合されています。Visual Studio のプロジェクトには、複数のクラス (Grasshopperコンポーネント) を含めることができ、それらは1つのクラスライブラリにコンパイルされ、ユーザー間で簡単に配布することができます。

## 4.3. KUKA KRL COMPONENT。応用設計と評価

KUKA KRLコンポーネントのマスカスタマイゼーション機能は、様々なパラメトリックデザインでテストされました。一つのデザインタスクは、様々な開口部を持つ個々のスタッキング可能な要素から、パラメトリックなEPSレンガ壁を作成することでした。KUKA KRLの基本的なツールパスは、2つのパラメトリックな曲線で構成されており、形状や開口部が異なる、さまざまな物理的なスタッキング可能な3Dモジュールを作成します (図8)。図8のGrasshopperの定義では、様々なリアルタイムプレビューが可能で、設計者は設計と製造のプロセスを同時に追うことができます。パラメトリックなツールチップとガイドカーブの幾何学的な生成 (1)、製造される個々のモジュール (2)、スタックされた結果 (3)、自動ツールパスの生成 (4)、エンドエフェクタのシミュレーションが、すべて1つのビューポートに同時に表示されます。エンドエフェクタのシミュレーションとKRL出力(5)は、新しいカスタムKRLコンポーネントによって処理されます。これは、パラメトリック設計環境をカスタム設計ツールとして使用し、単一の幾何学的形状に変更を加えるという、我々の初期のアイデアを示すものです。

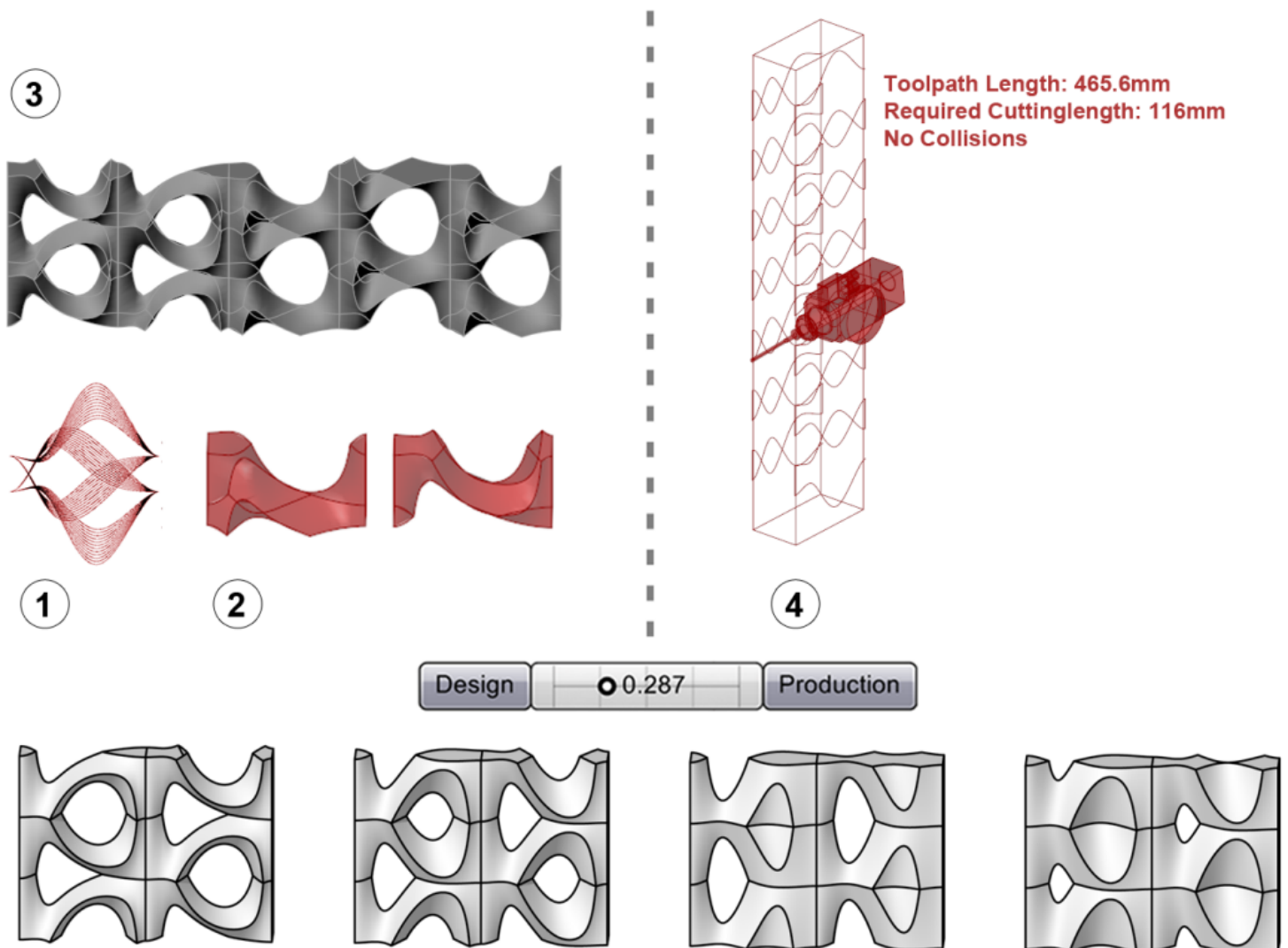


図8. 入力された形状のリアルタイムプレビュー  
(1): 個々のモジュール

- (2): 最終デザイン
- (3): フィージビリティ分析を伴うミリングシミュレーション
- (4): KRLコード
- (5): を同時に表示したプロダクション・インマネント・パラメトリック・モデル

従来のCAMソフトウェアとは対照的に、このモジュラーアプローチは、オープンでフレキシブルなパラメトリックCAM環境を可能にします。

Grasshopperに統合されたKUKAプラグインと従来のCAD/CAMを比較するために、同じパラメトリックウォールデザインの2つのプロトタイプをフライス加工しました。両方の結果の物理モデルは区別がつかず、市販のCAMソフトウェアでの準備には、KRLコンポーネントに比べてわずかに時間がかっただけでした。KRLコンポーネントはKRLコードを瞬時に生成し続けることができるのに対し、CAD/CAMソフトウェアはその都度セットアップを行う必要があるため、バリエーションが続くと時間差が広がってしまいます。図8のデザインバリエーションでは、ツールパスは1500ポイントで構成されています。KRLファイルは2GHzのノートパソコンで45ms以内に生成され、理論的には1秒間に20個のデザインバリエーションを自動出力することができる。この迅速な対応により、設計者は、CAMソフトウェアで各ファイルを手動で処理したり、ロボットのキネマティクスのために後処理をしたりすることなく、デザインバリエーションを素早く出力して製作することができます。さらに、計算は設計プロセスと並行して行われるという利点もあります。設計者が曲線やスタック可能な3Dモジュールなどの幾何学的なエンティティを「遊ぶ」ことができる一方で、KUKAプラグインは製作の実現性や起こりうる衝突についてフィードバックを提供します。

Grasshopperに統合されたKUKAプラグインをフルスケールの建築プロトタイプで評価するために、2009/2010年に行われたロボットファブリケーションプロジェクト「Plug & Play」を再考しました。このプロジェクトでは、子供の遊び場デザインの一環として、12m<sup>2</sup>の自由形状の内面に18枚のユニークな自由形状パネルを製作する必要がありました (図9)。KUKAプラグインを使用して、粗加工、仕上げ加工、側面加工などのさまざまな加工方法を評価し、成功させました。



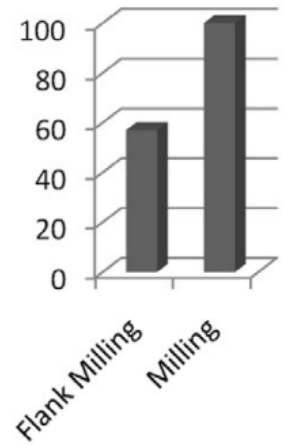
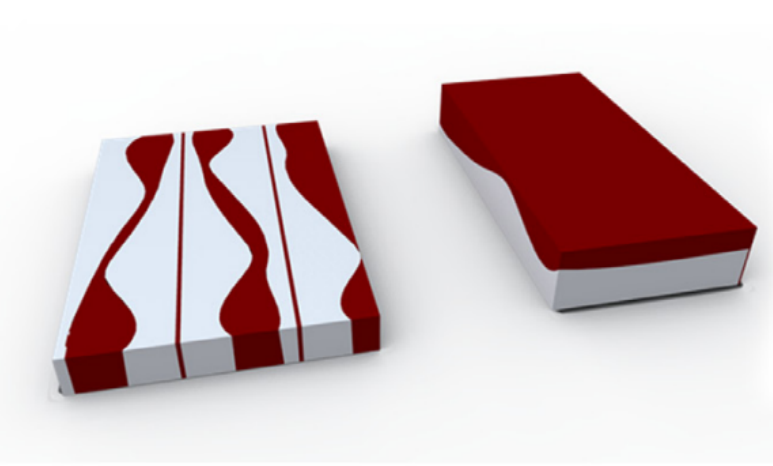


図9.

(左): 12m<sup>2</sup>の自由曲面を18枚のパネルに分割

(中、右): 材料効率の観点から側面フライスとポイントフライスを比較

粗加工と仕上げ加工は、一般的なポイントミル加工であり、エンドミルの刃先を使用してストック材を1層ずつ粗く削り (粗加工)、最終的な表面仕上げを行う (仕上げ加工)。一方、フランクミーリング (Schindler 2009) は、エンドミルの全長を利用して、ストックモデルから仕上げ面を切り出す方法です。

ポイントミリングとフランクミリングではこのような本質的な違いがあるため、パネルの形状を作るために2つの異なるアプローチが開発されました。粗面化・仕上げ加工では、XPSのブロックをいくつか使って最終的な表面を加工しました。側面フライス加工では、自由形状を罫線で近似し、各面の幅を共通のXPSパネルの高さに対応させました。その後、1回の切削で仕上げ面を作成した。

計算的にも物理的にも、このプロジェクトではフランク・ミーリングが最適なソリューションであることがわかりました。ポイントミル加工と比較して、フランクミル加工では、ツールパス長 (機械時間) が約80%短縮され、材料費も40%以上削減されました (図9)。また、Waldt (2005) も同様の結果を報告しています (フランクミル加工)。

Rapid Traverse : 13 m  
Toolpath Length: 224 m  
Best Case Scenario: 16 min

Rapid Traverse: 2.5 m  
Toolpath Length: 16 m  
Best Case Scenario: 1:20 min

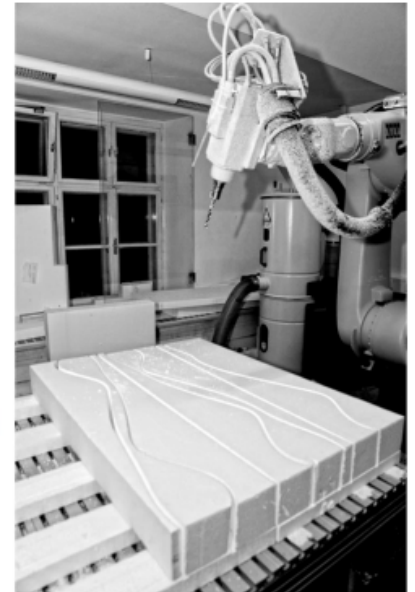
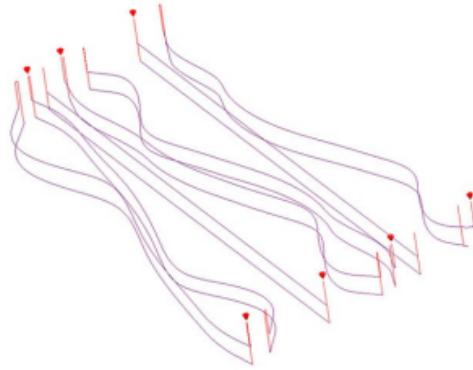
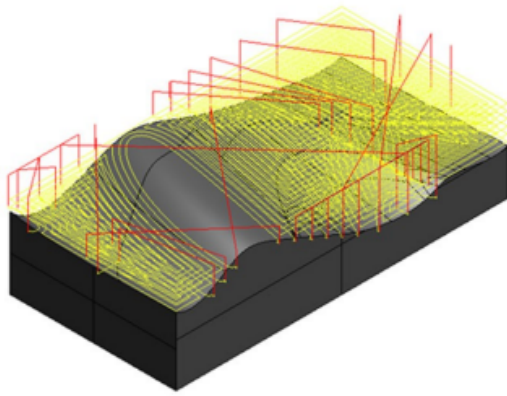


図10. 自由形状のプレイグラウンド: ポイントとフランクのミリング戦略の比較

図10は、市販のCAMソフトウェアで提案したパネルのミーリング手法を比較し、フランクミーリングがツールパス長とマシンタイムに与える影響を示したものです。

ポイントミルのツールパス長が長くなると、データ量が多すぎてロボットコードの生成が著しく遅くなるため、パラメトリックデザイン環境にも影響を与えます。自由形状のプレイグラウンドデザインでは、フランクミル加工のみが、リアルタイムに近い流麗なデザイン環境での作業を可能にします。そのため、パラメトリックな設計環境とマスカスタマイゼーションの効率化のためには、ミリングストラテジーの正しい選択が不可欠です。そのため、設計者はパラメトリックモデルを作成する前に、ミリング手法を評価する必要があります。設計から製造までを自動化するには、このような判断を可能にするミリングの経験が必要です。

本稿で紹介するパラメトリックウォールと子供の遊び場のデザインという2つの実現例は、CAD/CAM機能を統合したパラメトリックデザインソフトウェアが、従来のCAMソフトウェアツールだけでは実現できなかったカスタマイズされた最適な製作戦略に対応できることを示しています。単に最終的な3D CAD ファイルを製作工場に送るのではなく、製作のためにデザインを最適化することで、加工時間と材料費の大幅な削減、さらには廃棄物の削減というエコロジーな側面も実現できます。

Grasshopper用のKUKAコンポーネントは、最適化された設計からロボットによる製造までの流動的なワークフローを可能にします

## 5. 結論と展望

日本では1980年代から、産業用ロボットが手作業に代わるものとして本格的な建設に適用されてきたが、自動化の難しさと経済的な理由からあまり成功していない。Bechthold (2010) が述べているように、「カスタマイズされた建設」にロボットを使用するためには、自動化されたプログラミング戦略を開発することが基本となる。私たちの研究の目的は、幅広い建築家が、従来のCNCマシンと同様の使いやすさと、マスカスタマイゼーションに向けた直感的なアプローチでロボットを使用できるようにすることです。**この論文で発表されたパラメトリックインターフェースは、産業用ロボットが幅広い設計者やCADユーザーによって本格的な建設技術として使用されるための第一歩となります。**建築デザインの観点から、私たちは、必要なすべての制約がパラメトリックモデルに統合されているロボット製造デザインで「遊び」たいと思っています。GramazioとKohlerのロボットによるレンガ積み  
の壁に比べて、KUKA|prcのコンポーネントをミリング用にデザインするという私たちのアプローチは、もちろん幾何学的にはそれほど複雑な問題ではありませんが、デザインにおけるある種の遊び心と、KUKAロボットのような技術的に洗練されたCNCマシンを使用するためのインターフェースの作成の容易さを示しています。Grasshopperの多機能性は、産業用ロボットの多機能性と比較することができます。デザイナーは、ロボットミリング用のフライスカッターや、レンガ積み用のグリッピングツールなど、さまざまなツールをプラグアンドプレイで接続することができます (図11)。これにより、本稿で紹介するKUKA|prcコンポーネントは、異なる幾何学的ソリューションやツール制約を簡単にプラグアンドプレイすることで、多種多様なロボットアプリケーションを実現するように変更可能であり、産業用ロボットをCAADに統合し、これまでのCAD/CAMを超えていくことができると期待できます (図12, 13)。

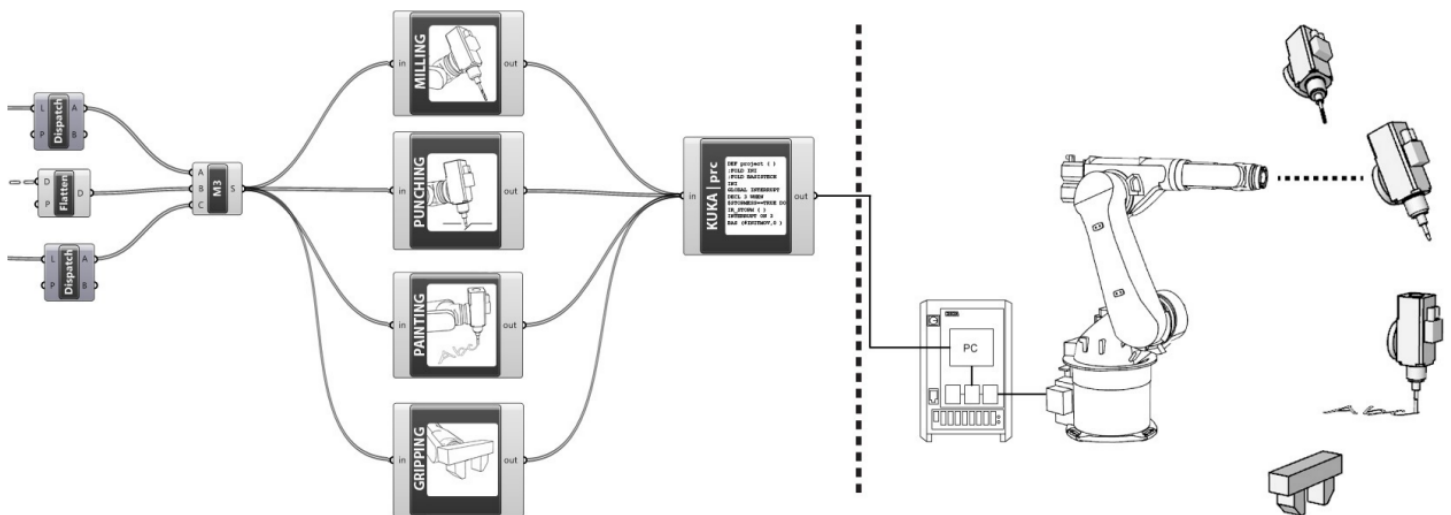


図11. KUKA|prcを用いたモジュール式パラメトリックロボット制御