卒業論文

Xtion Pro Live カメラを用いた 複数移動ロボットのビジュアルフィードバック制御の基礎実験計

F112026　三木　康平

指導教員　永田寅臣　教授

2016 年 3 月

山口東京理科大学 工学部 機械工学科

概要

目次

第 1 章

緒言

現在の商品生産方式はこれまでと大きく変わってきている．高度経済成長期には, 大量生産で製造コストを下げ, 一般に普及させることが重要であった．このような少品種大量生産は, ラインの流れ作業による製造を生み, 同時に増加した繰り返し作業をロボットが代行することで自動化が図られた．一方, 現在では消費者嗜好の多様化に合わせて変種変量生産が求められることが多くなってきた．要求が常に変動するような作業は, ロボットを変化に応じて環境整備・作業教示 ・ 調整をする必要が生じるが, これでは採算が合わないため, 依然として工場制手工業が多く残る．したがって, 人のように状況判断を行い, 幅のある業務をこなすことができるような汎用機械としてのロボットが求められるようになってきている．

これまで, 永田研究室では, CL データを用いた多関節ロボットによる発泡スチロールの加工実験や, CLS データに基づき教育用ロボットを動作させる CAD/CAM インタフェイスの提案を行ってきた [1, 2].

第 2 章

運動学

ロボットアームの手先を目的の位置に移動させたいとき, 各関節角度と関節間の長さ現在の手先位置が分かっていれば, 運動学を用いて必要な姿勢を計算することが出来る. 運動学とは, ロボットアームの関節角度や関節間の長さと手先位置・姿勢の関係を数式で表したものであり, 各関節角度から手先位置・姿勢を求めることを順運動学, 手先位置・姿勢から各関節角度を求めることを逆運動学という. また, ロボットアームは各関節ごとに座標系（関節座標系）を設定することができ, この座標系と台座部に設定した基準座標系は運動学に用いることで互いに変換することができる.

2.1 順運動学

順運動学とは, ロボットアームの各関節角度から手先位置・姿勢を求めること, すなわち手先位置の関節座標系から基準座標系への変換ということができる. ここでは関節座標の手先位置を基準座標系に変換する方法について示す. まず簡単のために図 2.1 のような系の並進移動について考える.

図 2.1. Polishing scene using a conventional industrial robot with a servo spindle.

移動後の原点 $O'(x'_0, y'_0, z'_0)$ は, 移動前の原点 $O(x_0, y_0, z_0)$ から見ると次のように表される.

$$\begin{aligned} x'_0 &= x_0 + x_1 \\ y'_0 &= y_0 + y_1 \\ z'_0 &= z_0 + z_1 \end{aligned}$$

上式をベクトル $\boldsymbol{r}$ を用いて表すと, 以下のようになる.

$$O' = O + \boldsymbol{r} \tag{2.1}$$

関節座標系から基準座標系への変換を行うには、並進移動だけでなく回転移動についても考える必要がある. そこで図 2.2 のような簡単な系の回転移動について考える.

図 2.2. Polishing scene using a conventional industrial robot with a servo spindle.

まずベクトル $\boldsymbol{P}(x_1, y_1, z_1)$ は, 各軸方向の単位ベクトル $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ を用いて次のように表される.

$$\boldsymbol{P} = x_1\boldsymbol{i} + y_1\boldsymbol{j} + z_1\boldsymbol{k} \tag{2.2}$$

次に, Z 軸を基準としてベクトル $\boldsymbol{P}$ を含む X-Y 平面を θ だけ回転した座標系を新しく X', Y', Z' 座標系とすると, その座標系におけるとすると, ベクトル $\boldsymbol{P'}(x_2, y_2, z_2)$ は単位ベクトル $\boldsymbol{i'}, \boldsymbol{j'}, \boldsymbol{k'}$ を用いて次のように表される.

$$\boldsymbol{P'} = x_2\boldsymbol{i'} + y_2\boldsymbol{j'} + z_2\boldsymbol{k'} \tag{2.3}$$

ここで, 単位ベクトル単位ベクトル $\boldsymbol{i'}, \boldsymbol{j'}, \boldsymbol{k'}$ は回転前の座標系から見ると

$$\begin{bmatrix} \boldsymbol{i'} \\ \boldsymbol{j'} \\ \boldsymbol{k'} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{j} \\ \boldsymbol{k} \end{bmatrix} \tag{2.4}$$

と表せる. さらに式 (2.4) に式 (2.3) と式 (2.2) を代入すると

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

さらに見通しをよくするため, 単位ベクトルを省略し以下のように表す.

$$\boldsymbol{P}' = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{P} \tag{2.5}$$

すなわち Z 軸を基準としたベクトルの回転移動は, 移動前後の位置移動および相対角 θ を用いて表すことができ, これは X 軸, Y 軸においても成り立つ. この式 (2.5) 中央の行列を回転行列といい R で表すこととする. 並進移動と回転移動とを用いることにより, 図 2.3 に示すような一見複雑な座標系の変換も次のような式で表すことができる.

図 2.3. Polishing scene using a conventional industrial robot with a servo spindle.

$$^{A}\boldsymbol{P} = {}^{A}\boldsymbol{R}_{B}{}^{B}\boldsymbol{P} + {}^{A}\boldsymbol{r}_{B} \tag{2.6}$$

ここで ${}^{A}\boldsymbol{R}_{B}$ は座標系 A から座標系 B への回転行列である. この関係を各関節ごとに用いることで, 手先位置を基準座標系からの位置として求めることができる.

2.2　逆運動学

順運動学が座標変換を行うことで解が 1 つ求まるのに対して, 逆運動学では図 2.4 の場合のように手先位置が同じ位置にある場合でも各リンクの位置と姿勢にはいくつかの解が存在することや, 解自体が存在しないこともある. このような問題があるため, 一般に順運動学より逆運動学の方が問題を解くのが難しい.

逆運動学でアームの関節角度を計算する例として, 図 2.5 のような場合を考える. まず, 角度 α と $|\boldsymbol{OP}|$ の関係は余弦定理を用いて次式で表される.

$$\cos\alpha = \frac{L_1^2 + L_2^2 - |\boldsymbol{OP}|}{2L_1L_2} \tag{2.7}$$

図 2.4. Polishing scene using a conventional industrial robot with a servo spindle.

図 2.5. Polishing scene using a conventional industrial robot with a servo spindle.

式 (2.7) より角度 α は $\alpha = \cos^{-1}(\cos\alpha)$ と表せるので, 結果的に θ_2 は次式で表される.

$$\theta_2 = \pi - \alpha \tag{2.8}$$

同様に, 余弦定理を用いて角度 β と $|OP|$ の関係は次式で表され,

$$\cos\beta = \frac{|\boldsymbol{OP}|^2 + L_1^2 - L_2^2}{2|\boldsymbol{OP}|L_2}$$

$\beta = \cos^{-1}(\cos\beta)$ である. 最後に角度 γ は, 補助線 $|\boldsymbol{OP}|$ と手先位置の x 座標との関係から

$$\cos\gamma = \frac{x}{|\boldsymbol{OP}|}$$

であるので, $\gamma = \cos^{-1}(\cos\gamma)$ である. よって, 角度 β および角度 α を用いて θ_1 は次式で表される.

$$\theta_1 = \gamma - \beta \tag{2.9}$$

よって, 手先位置から関節角度 θ_1, $theta_2$ を求めることができた.

また別の方法として, 図 2.6 のような場合を考える. この場合, 手先位置は次式で表される.

$$\begin{aligned} x_p &= l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ y_p &= l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \end{aligned} \tag{2.10}$$

図 2.6. Polishing scene using a conventional industrial robot with a servo spindle.

式 (2.10) の両辺を時間で微分し, 行列で表すと

$$\frac{d\boldsymbol{P}}{dt} = \boldsymbol{J}(\boldsymbol{\theta})\frac{d\boldsymbol{\theta}}{dt} \tag{2.11}$$

ここで

$$\boldsymbol{P} = [x_p, y_p]^{\mathrm{T}}, \boldsymbol{\theta} = [\theta_1, \theta_2]^{\mathrm{T}}, \boldsymbol{J}(\theta) = \begin{bmatrix} -(l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

である. 式 (2.12) は手先位置の速度と各関節における角速度の関係を表しており, 両辺の変換を担う行列 $\boldsymbol{J}$ を一般にヤコビ行列という. さて, 逆運動学は手先位置から各関節角度を求める問題のことであったので, 式 (2.12) を以下のように変形する.

$$\frac{d\boldsymbol{\theta}}{dt} = \boldsymbol{J}^{-1}\frac{d\boldsymbol{P}}{dt} \tag{2.12}$$

しかし, $\boldsymbol{J}^{-1}$ は必ずしも存在しない. $\boldsymbol{J}^{-1}$ が存在しない条件は, $\det\boldsymbol{J}^{-1}$ で求めることができ, その時の姿勢を特異姿勢もしくは特異点という.

第 3 章

人工知能

3.1　人工知能

人工知能 (artificial intellgence:AI) は, 1947 年にアラン・チューリングによって提唱された概念であり, その定義は正確に定められておらず, 専門家によっても意見が異なる. 例えば, 次のような意見がある.

栗原聡	電機通信大学	人工的につくられる知能であるが, その知能のレベルは人を越えているものを想像している
山川宏	ドワンゴ人工知能研究所	計算機知能のうちで, 人間が直接・関節に設計する場合を人工知能と呼んで良いのではないかと思う
松尾豊	東京大学	人工的につくられた人間のような知能, ないしはそれをつくる技術. 人間のように知的であるとは, 「気づくことのできる」コンピュータ, つまり, データの中から特徴量を生成し現象をモデル化することのできるコンピュータという意味である

また, 定義が曖昧であるにも関わらず日常には AI という言葉があふれている. この原因として, 新井は文献 [3] にて,「AI」と「AI 技術」という言葉が混同して使われている点を指摘している. 同文献によると, AI 技術というのは, AI を実現するために開発されている様々な技術（音声認識技術, 自然言語処理技術, 画像処理技術など）であり, AI は AI 技術開発の先にあるゴールである. 以後は, この文献に従い AI を実現するために開発されている技術, AI 技術について説明していく.

3.2　機械学習

3.3　ニューラルネットワーク

第 4 章

実機による動作実験

4.1 Dobot

TechShare 社が販売している Dobot Magician(Dobot) は, 4 自由度のロボットアームでありエンドエフェクタを取り換えることによって 3D プリンタのような積載加工, エンドミルによる切削加工ペンツールを使った印字などが行える．Dobot の質量と可搬質量はそれぞれ 3.4kg と 500g であり, 位置繰り返し精度は 0.2mm と教育ロボットとしては優れている．また Dobot 社より, C や MFC(Microsoft Foundation Class), Python など複数のプログラミング言語で API が提供されており, 個人でプログラムを組む際の敷居も低い．今回は, その中の Python API を統合開発環境である Visual Studio2019 上に実装し, Dobot 本体を動作させるプログラムの開発を行った．API とはあるコンピュータプログラム（ソフトウェア）の機能や管理するデータなどを、外部の他のプログラムから呼び出して利用するための手順やデータ形式などを定めた規約のことである．今回はその中から Dobot 本体の姿勢を取得するための GetPose() また, Python に標準で搭載されている GUI 作成キットである Tkinter を用いて, 開発した動作プログラムを簡単に扱えるよう Dobot 制御用のユーザインタフェースの作成も行った.

4.2 学習データ

4.3 実験結果

第 5 章

今後の方針

謝辞

本研究は，山口東京理科大学大学院基礎工学研究科基礎工学専攻で行われたものである．

参考文献

[1] 大塚　章正, 永田　寅臣, 中村　航輔, “CL データの位置姿勢情報に基づく軌道追従制御器を用いた発泡スチロール加工ロボット”, ロボティクス・メカトロニクス講演会 2014 講演概要集, 3P1-L01, pp.1-3, 2014.

[2] 鈴木真太朗，永田寅臣，渡辺桂吾，“教育用ロボット DOBOT のための CAD/CAM インタフェイス”，ロボティクス・メカトロニクス講演会 2018 講演論文集, 2P1-L07(1-2), 北九州国際会議場, 2018.

[3] 新井紀子 『AI vs. 教科書が読めない子供たち』（東洋経済新報社, 2018）