

移動距離と全身の運動を一つの系として記述可能な ロボット運動学の定式化手法

関口 叡範 東京都立大学

Sekiguchi Masanori Tokyo Metropolitan University

1. 緒 言

少子高齢化による労働人口の減少が続く近年において、ロボットの活躍は益々期待されている。そこで本論文では、ロボットの制御における核心技術の一つである運動学（キネマティクス）について議論する。ロボットの運動学は、近年注目を集めている機械学習の性能にも大きく影響を及ぼす。例えば、逆運動学を計算可能な場合と、そうでない場合では、学習方式あるいは学習効率に大きな差が生じる。

運動学はロボット工学における基本的な概念の一つであり、古くから多くの文献で取り上げられている。大抵の場合、順運動学の計算には Denavit-Hartenberg の表記法 (DH 法) [1] が用いられ、微分運動学 (ヤコビ行列) の定式化には偏微分が用いられる。これらの方法は、ロボット工学が一つの学問分野として認知され始めた 1980 年代から既に使用されており、それが現在のロボット工学にまで引き継がれている。1980 年代では、多関節ロボットの主な用途が画一的な単純作業に限定されていたことから、自由度が 4～6 程度の簡素なロボットが主流となっていた。しかし近年では、移動マニピュレータや腰に自由度を持った双腕型のロボットなどが普及し、ロボットの構造や自由度が大きく多様化・冗長化した。これにより、順運動学やヤコビ行列の計算過程に従来から潜在していた問題点が、より顕著に現れるようになった。

DH 法における問題点の一つは、隣接する二つの関節軸が互いに平行であった場合、DH パラメータの一つが不定となることである [2]。この問題は DH 法における特異点問題として古くから知られており、既に多くの解決策が提案されている [3]～[9]。しかし、DH 法には特異点の問題だけでなく、各関節の位置・姿勢が計算結果に残らないことや、DH パラメータの計算手順が複雑であるといった実用上の問題点があることも指摘されている [10]。これらの問題点に加え、ロボットの構造が多様化した近年においては、理論の表記に関する他の問題点も顕在化している。例えば、双腕型のロボットを対象とした場合、右腕か左腕かを判別するための添え字を各変数に追加する必要があるが、従来の定式化手法では既に多くの添え字が使用されており、数式として表現することが困難である。

微分運動学の計算過程においても二つの問題点がある。一つ目は、ロボットの自由度が大きくなるにつれて、ヤコビ行列の計算式が複雑化し、実装コストや計算時間の点で

不都合をもたらすことである。場合によっては、ヤコビ行列の計算式が A4 用紙数十枚分に相当する長さにも肥大化することもある。二つ目は、習慣的にヤコビ行列として扱われている行列が、微分積分学におけるヤコビ行列の定義に順守していないことである。この問題点の原因は、手先効果器の姿勢を表すベクトル (オイラー角など) を関節変位ベクトルで偏微分しても、角速度に関する微分運動学は導出できないことにある [11]。偏微分によって定義される本来のヤコビ行列と、そうでないものが区別されていないという点で、ヤコビ行列という名称を使用することは厳密には適切ではない。この問題点を考慮した一部の文献では、基礎ヤコビ行列 (the basic Jacobian matrix) [12] [13] や幾何ヤコビ行列 (geometric Jacobian matrix) [14]～[16] といった他の名称が使用されている。しかし、名前にヤコビ行列という名称が含まれているため、誤解の余地を与え得るという問題点は依然として残されている。

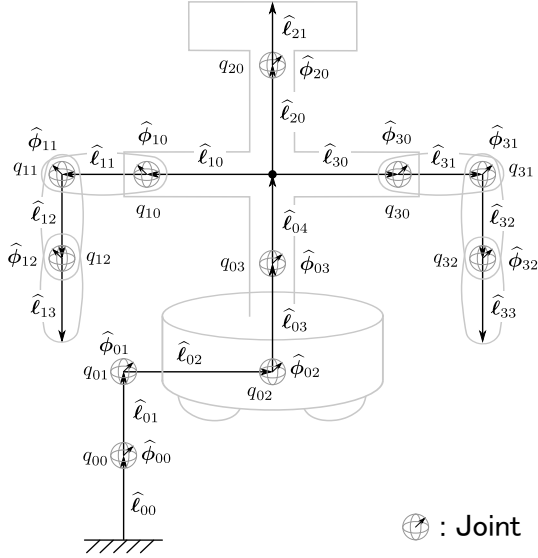
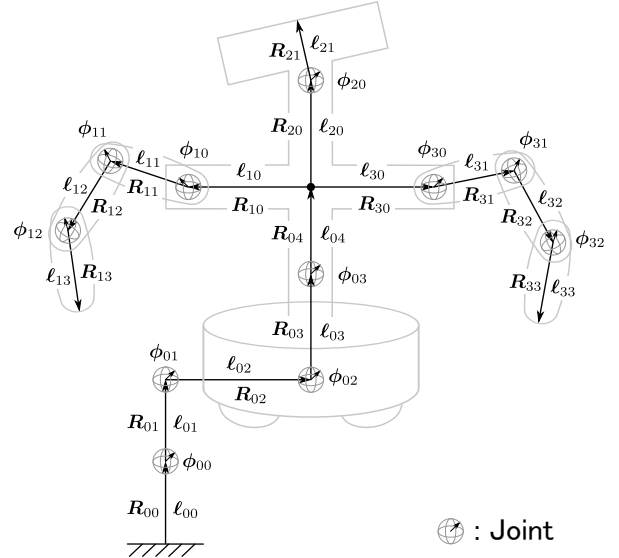
そこで本論文では、順運動学とヤコビ行列の基本を見直し、前述した問題点を解決するための新たな定式化手法を提案する。まず、第 2 章と第 3 章では、それぞれ順運動学と微分運動学 (ヤコビ行列) の定式化手法について述べる。本論文で提案する定式化手法を用いることで、分岐点や直進対偶、移動能力を有するロボットにおいても、同一の手順で運動学を計算することが可能となる。第 4 章では、提案した運動学の定式化手法を逆運動学の数値解法に適用した事例を示す。第 5 章では、双腕型の移動マニピュレータを用いたシミュレーション実験を行い、提案手法の具体的な応用例を示すと共に、提案手法に含まれる各計算式の妥当性を検証する。第 6 章の結言では、まとめと今後の課題について述べる。

2. 順 運 動 学

本章では、本論文で提案する順運動学の定式化手法について説明した後、提案手法を DH 法と比較したときの利点と欠点について述べる。

2.1 順運動学の定式化

大抵の場合、多関節ロボットの機械的な構造は、分岐点を有する開リンク機構の一種として扱うことができる。人型ロボットのように、ロボットが三次元空間上を任意に移動する場合においても、ロボットの位置と姿勢をそれぞれ 3 つの直進対偶と 1 つの球対偶を用いて表現することで、運動学モデルを開リンク機構の一種として扱うことができ

図1 基準姿勢 ($q = \mathbf{0}$) のときの運動学パラメータ図2 任意の q に対する運動学パラメータ

る [17]. そこで本節では、一つの幹と複数の枝によって構成されている樹状型の剛体リンク系を対象に、その順運動学について考える。幹と各枝はそれぞれ直鎖状のリンク連鎖であるとし、それらが一点で連結することによって、リンク系が構成されているものとする。

枝の本数を r とし、幹の自由度を n_0 、各枝の自由度を順に n_1, n_2, \dots, n_r とする。図1のリンク系では、 $r = 3$ 、 $n_0 = 4$ 、 $n_1 = 3$ 、 $n_2 = 1$ 、 $n_3 = 3$ である。リンク系全体の自由度は $n := n_0 + n_1 + \dots + n_r$ となる。リンク系全体の関節変位ベクトル q を次のように定義する。

$$q = \begin{bmatrix} q_0^T & q_1^T & \dots & q_r^T \end{bmatrix}^T \in \mathbb{R}^n \quad (1)$$

ここで、 q_0 は幹の各関節変位を表し、 q_1, q_2, \dots, q_r は各枝の各関節変位を表す。また、幹と枝の両方をまとめて直鎖リンクと呼ぶことにし、その番号を i とする。直鎖リンク i における j 番関節の変位 (q_i の第 j 成分) を q_{ij} とする。本章では、以降でも i は直鎖リンクの番号を表し、 j は関節またはリンクの番号を表す。

まず、リンク系が基準姿勢 ($q = \mathbf{0}$) のときにおける運動学パラメータを定義する。リンクベクトル $\hat{l}_{ij} \in \mathbb{R}^3$ と関節軸ベクトル $\hat{\phi}_{ij} \in \mathbb{R}^3$ を図1のように定める。リンクベクトル \hat{l}_{ij} は隣接する関節の相対的な位置関係を表すベクトルである。関節軸ベクトル $\hat{\phi}_{ij}$ は、関節 j が回転対偶であれば単位回転軸ベクトルを表し、直進対偶であれば単位方向ベクトルを表す。 \hat{l}_{ij} と $\hat{\phi}_{ij}$ はリンク系の構造を表す定数のベクトルであり、既知の変数となる。

図2のように、任意の q に対する各リンクの姿勢を表す回転行列を R_{ij} とする。回転行列 R_{ij} は関節変位ベクトル q によって生じる姿勢の変化を表しており、 $q = \mathbf{0}$ のときは全ての R_{ij} が単位行列となる。 $j+1$ 番リンクの姿勢を表す回転行列は、 j 番関節の種類 (回転対偶または直進対偶) に応じて次のように表される。

$$R_{i,j+1} = \begin{cases} R_{ij} R(q_{ij}, \hat{\phi}_{ij}) & (\text{回転対偶}) \\ R_{ij} & (\text{直進対偶}) \end{cases} \quad (2)$$

ここで、 $R(q_{ij}, \hat{\phi}_{ij})$ は回転角度 q_{ij} と回転軸ベクトル $\hat{\phi}_{ij}$ による回転変換を回転行列形式で表したものである (付録A参照)。ただし、各直鎖リンクの0番リンクにおける回転行列 R_{i0} は、上式では計算できないため次式を用いる。

$$R_{i0} = \begin{cases} I & (i = 0) \\ R_{0n_0} & (i \neq 0) \end{cases} \quad (3)$$

ここで、 I は単位行列である。擬似コードで回転行列の計算手順を表すと以下ようになる。

Algorithm 1

 Procedure for computing the R_{ij} .

- 1: for $i = 0, 1, \dots, r$ do
 - 2: $R_{i0} = \begin{cases} I & (i = 0) \\ R_{0n_0} & (i \neq 0) \end{cases}$
 - 3: for $j = 0, 1, \dots, n_i - 1$ do
 - 4: $R_{i,j+1} = \begin{cases} R_{ij} R(q_{ij}, \hat{\phi}_{ij}) & (\text{回転対偶}) \\ R_{ij} & (\text{直進対偶}) \end{cases}$
 - 5: end for
 - 6: end for
-

回転行列 R_{ij} が既知となることで、任意の q に対する各リンクベクトル $l_{ij} \in \mathbb{R}^3$ と各関節軸ベクトル $\phi_{ij} \in \mathbb{R}^3$ もそれぞれ次式によって計算可能となる。

$$l_{ij} = R_{ij} \hat{l}_{ij} \quad (4)$$

$$\phi_{ij} = R_{ij} \hat{\phi}_{ij} \quad (5)$$

擬似コードで l_{ij} と ϕ_{ij} の計算手順を表すと次のようになる。

Algorithm 2
 Procedure for computing the ϕ_{ij} and ℓ_{ij} .

```

1: for  $i = 0, 1, \dots, r$  do
2:   for  $j = 0, 1, \dots, n_i - 1$  do
3:      $\phi_{ij} = \mathbf{R}_{ij} \hat{\phi}_{ij}$ 
4:      $\ell_{ij} = \mathbf{R}_{ij} \hat{\ell}_{ij}$ 
5:   end for
6:    $\ell_{in_i} = \mathbf{R}_{in_i} \hat{\ell}_{in_i}$ 
7: end for
  
```

次に、絶対座標系（ベース）に対する各関節の位置ベクトルを $\mathbf{p}_{ij} \in \mathbb{R}^3$ とする。ただし、 $j = n_i$ のときのみは \mathbf{p}_{ij} は関節の位置ではなく、各直鎖リンク（幹または枝）の終端位置となる。それぞれの位置ベクトルは、 j 番関節の種類（回転対偶または直進対偶）に応じて次のように表される。

$$\mathbf{p}_{i0} = \begin{cases} \ell_{00} & (i = 0) \\ \ell_{i0} + \mathbf{p}_{0n_0} & (i \neq 0) \end{cases} \quad (6)$$

$$\mathbf{p}_{i,j+1} = \begin{cases} \mathbf{p}_{ij} + \ell_{i,j+1} & (\text{回転対偶}) \\ \mathbf{p}_{ij} + \ell_{i,j+1} + q_{ij} \phi_{ij} & (\text{直進対偶}) \end{cases} \quad (7)$$

j 番関節が直進対偶のときにおける関節間の相対的な位置関係を図 3 に示す。

擬似コードで位置ベクトルの計算手順を表すと次のようになる。

Algorithm 3
 Procedure for computing the \mathbf{p}_{ij} .

```

1: for  $i = 0, 1, \dots, r$  do
2:    $\mathbf{p}_{i0} = \begin{cases} \ell_{00} & (i = 0) \\ \ell_{i0} + \mathbf{p}_{0n_0} & (i \neq 0) \end{cases}$ 
3:   for  $j = 0, 1, \dots, n_i - 1$  do
4:      $\mathbf{p}_{i,j+1} = \begin{cases} \mathbf{p}_{ij} + \ell_{i,j+1} & (\text{回転対偶}) \\ \mathbf{p}_{ij} + \ell_{i,j+1} + q_{ij} \phi_{ij} & (\text{直進対偶}) \end{cases}$ 
5:   end for
6: end for
  
```

順運動学を計算する過程で必要となった全ての変数を図 4 に示す。例えば、これらの変数を C 言語で実装する場合、一つのブロックをクラスとして定義し、それを二次元配列にすることなどが考えられる。

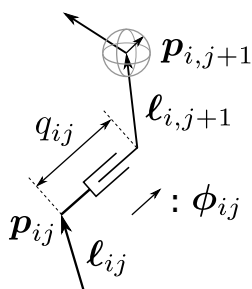


図 3 j 番関節が直進対偶のときにおける関節間の相対的な位置関係

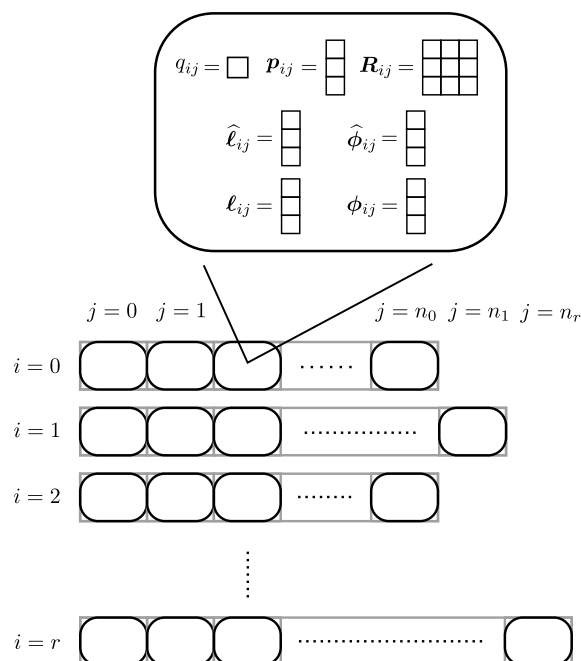


図 4 順運動学の計算に使用した変数

2.2 DH 法と比較したときの利点と欠点

DH 法と比較したとき、前述した順運動学の定式化手法には以下のような利点がある。

- (1) プログラミング言語との互換性が高く、実装が容易であること。
- (2) 分岐点や直進対偶を含むロボットにおいても、同一の手順で順運動学を計算可能であること。
- (3) 分岐点を有するロボットを対象とした場合でも、添え字に含まれる変数の数は基本的に二つ以下に収まること。
- (4) リンクベクトルや関節軸ベクトルの向きが、絶対座標系の座標軸と平行でない場合でも適用可能であること。
- (5) 絶対座標系に対する各関節の位置と各リンクの姿勢が明確となるため、計算結果を干渉チェックやヤコビ行列の計算に再利用できること。

一方で、提案した順運動学の定式化手法には、DH 法よりも多くの変数を計算の過程で要するという欠点がある。

3. 微分運動学

本章では、分岐点や直進対偶を含む多関節ロボットを対象とした微分運動学の定式化手法について述べる。本章で述べる定式化手法には三つの利点がある。一つ目は、計算過程に偏微分が含まれないため、実装が容易となる点である。二つ目は、第 2 章で述べた順運動学の計算結果が再利用されるため、新たに必要となる変数や演算処理が少ないことである。三つ目は、ロボットの構造に関わらず、同じ理論・同じプログラムで微分運動学を計算できることである。

各効果器（幹と各枝の終端位置）の速度と角速度を表すベクトルを次のように定める。

$$\mathbf{v}(\mathbf{q}) = \left[\mathbf{v}_0^T(\mathbf{q}) \quad \mathbf{v}_1^T(\mathbf{q}) \quad \dots \quad \mathbf{v}_r^T(\mathbf{q}) \right]^T \in \mathbb{R}^{6(r+1)} \quad (8)$$

ここで、 $\mathbf{v}_0(\mathbf{q}) \in \mathbb{R}^6$ は幹の終端位置（分岐点）における速度と角速度を表し、 $\mathbf{v}_1(\mathbf{q}), \mathbf{v}_2(\mathbf{q}), \dots, \mathbf{v}_r(\mathbf{q}) \in \mathbb{R}^6$ は各枝の終端位置における速度と角速度を表す。

以降では、次式を満たすような行列 $\mathbf{J}(\mathbf{q})$ を求めることを考える。

$$\mathbf{v}(\mathbf{q}) = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (9)$$

行列 $\mathbf{J}(\mathbf{q})$ は、習慣的にヤコビ行列と呼ばれている。しかし、特殊な機構を除き、行列 $\mathbf{J}(\mathbf{q})$ は多変数ベクトル関数の偏微分で求めることは不可能なため、厳密には行列 $\mathbf{J}(\mathbf{q})$ はヤコビ行列ではない。そこで本論文では、 $\mathbf{J}(\mathbf{q})$ を速度比行列（velocity ratio matrix）と呼ぶことにする。この名称は、行列 $\mathbf{J}(\mathbf{q})$ が各効果器と各関節の速度比を表していることに由来する。速度比行列 $\mathbf{J}(\mathbf{q})$ はブロック下三角行列となり、以下のようにブロック分割することができる。

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} {}^s\mathbf{J}_0 & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ {}^s\mathbf{J}_1 & {}^b\mathbf{J}_1 & \mathbf{O} & \cdots & \mathbf{O} \\ {}^s\mathbf{J}_2 & \mathbf{O} & {}^b\mathbf{J}_2 & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ {}^s\mathbf{J}_r & \mathbf{O} & \mathbf{O} & \cdots & {}^b\mathbf{J}_r \end{bmatrix} \quad (10)$$

ここで \mathbf{O} は零行列であり、各ブロックを表す変数では指数 (\mathbf{q}) が略記されている。さらに、行列 ${}^s\mathbf{J}_i \in \mathbb{R}^{6 \times n_0}$ および ${}^b\mathbf{J}_i \in \mathbb{R}^{6 \times n_i}$ を以下のように列ブロック分割する。

$${}^s\mathbf{J}_i = \begin{bmatrix} {}^s\mathbf{j}_{i0} & {}^s\mathbf{j}_{i1} & \cdots & {}^s\mathbf{j}_{i,n_0-1} \end{bmatrix} \in \mathbb{R}^{6 \times n_0} \quad (11)$$

$${}^b\mathbf{J}_i = \begin{bmatrix} {}^b\mathbf{j}_{i0} & {}^b\mathbf{j}_{i1} & \cdots & {}^b\mathbf{j}_{i,n_i-1} \end{bmatrix} \in \mathbb{R}^{6 \times n_i} \quad (12)$$

それぞれの列ベクトル ${}^s\mathbf{j}_{ij}$ と ${}^b\mathbf{j}_{ij}$ は、 j 番関節の種類（回転対偶または直進対偶）と状態（可動か否か）に応じて以下のように表される。

$${}^s\mathbf{j}_{ij} = \begin{cases} \begin{bmatrix} \phi_{0j} \times (\mathbf{p}_{in_i} - \mathbf{p}_{0j}) \\ \phi_{0j} \end{bmatrix} & \text{(回転対偶)} \\ \begin{bmatrix} \phi_{0j} \\ \mathbf{0} \end{bmatrix} & \text{(直進対偶)} \\ \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} & \text{(可動でない)} \end{cases} \quad (13)$$

$${}^b\mathbf{j}_{ij} = \begin{cases} \begin{bmatrix} \phi_{ij} \times (\mathbf{p}_{in_i} - \mathbf{p}_{ij}) \\ \phi_{ij} \end{bmatrix} & \text{(回転対偶)} \\ \begin{bmatrix} \phi_{ij} \\ \mathbf{0} \end{bmatrix} & \text{(直進対偶)} \\ \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} & \text{(可動でない)} \end{cases} \quad (14)$$

ここで、 $\mathbf{0} \in \mathbb{R}^3$ は零ベクトルである。関節変位が可動域の限界値に達しているときなどでは、関節は可動でなくなる。擬似コードで速度比行列の計算手順を表すと次のようになる。

Algorithm 4

Procedure for computing the ${}^s\mathbf{j}_{ij}$ and ${}^b\mathbf{j}_{ij}$.

```

1: for  $i = 0, 1, \dots, r$  do
2:   for  $j = 0, 1, \dots, n_0 - 1$  do
3:     Calculate the  ${}^s\mathbf{j}_{ij}$ .
4:   end for
5:   for  $j = 0, 1, \dots, n_i - 1$  do
6:     Calculate the  ${}^b\mathbf{j}_{ij}$ .
7:   end for
8: end for

```

ただし、上記の擬似コードでは ${}^s\mathbf{J}_0$ と ${}^b\mathbf{J}_0$ の両方が計算されるが、 ${}^s\mathbf{J}_0 = {}^b\mathbf{J}_0$ であり、実際にはどちらか一方の計算のみでも十分である。すなわち、式 (10) の ${}^s\mathbf{J}_0$ は ${}^b\mathbf{J}_0$ に置き換えても同じ計算結果となる。なお、 $i \neq 0$ のときは ${}^s\mathbf{J}_i \neq {}^b\mathbf{J}_i$ となる。また、 $r = 0$ のときは $\mathbf{J}(\mathbf{q}) = {}^s\mathbf{J}_0$ となり、このときの行列 $\mathbf{J}(\mathbf{q})$ は杉原の解説記事 [18] で述べられている基礎ヤコビ行列と等価になる。

4. 逆運動学への応用例

本章では、文献 [19] [20] で述べられている逆運動学の数値解法に対し、前述した順運動学と速度比行列を適用した事例を示す。提案手法を用いることで、ロボットに分岐点や直進対偶が含まれている場合においても、逆運動学の数値解法を一つの理論に定式化できることを示す。

4.1 誤差ベクトル

直鎖リンク i の終点における位置・姿勢をそれぞれ $\mathbf{p}_i(\mathbf{q}) \in \mathbb{R}^3$, $\mathbf{R}_i(\mathbf{q}) \in SO(3)$ とする。 $\mathbf{p}_i(\mathbf{q})$ は第 2 章の順運動学で求めた \mathbf{p}_{in_i} と一致し、 $\mathbf{R}_i(\mathbf{q})$ は $\mathbf{R}_{in_i} {}^h\mathbf{R}_i$ と一致する。ここで、 ${}^h\mathbf{R}_i$ はハンドの姿勢を表す定数の回転行列である。リンクベクトルの姿勢とハンドの姿勢は一致しているとは限らないため、リンクベクトルとハンドの相対的な姿勢関係を表す回転行列 ${}^h\mathbf{R}_i$ を導入した。また、直鎖リンク i の終点に対して与える目標の位置と姿勢をそれぞれ ${}^d\mathbf{p}_i \in \mathbb{R}^3$, ${}^d\mathbf{R}_i \in SO(3)$ とする。これらを用いて誤差ベクトル $\mathbf{e}(\mathbf{q})$ を次のように定める。

$$\mathbf{e}(\mathbf{q}) = \begin{bmatrix} \mathbf{e}_0^T(\mathbf{q}) & \mathbf{e}_1^T(\mathbf{q}) & \cdots & \mathbf{e}_r^T(\mathbf{q}) \end{bmatrix}^T \in \mathbb{R}^{6(r+1)} \quad (15)$$

$$\mathbf{e}_i(\mathbf{q}) = \begin{bmatrix} {}^d\mathbf{p}_i - \mathbf{p}_i(\mathbf{q}) \\ \phi({}^d\mathbf{R}_i \mathbf{R}_i^T(\mathbf{q})) \end{bmatrix} \in \mathbb{R}^6 \quad (16)$$

ここで、 $\phi(\ast)$ は回転行列を角軸ベクトル [21] に変換する関数である。

4.2 仮想バネと仮想トルク

誤差ベクトル $\mathbf{e}(\mathbf{q})$ に対して仮想的な直進バネと回転バネが張られていると考える。仮想直進バネは位置の誤差に対して張られ、仮想回転バネは姿勢の誤差に対して張られ

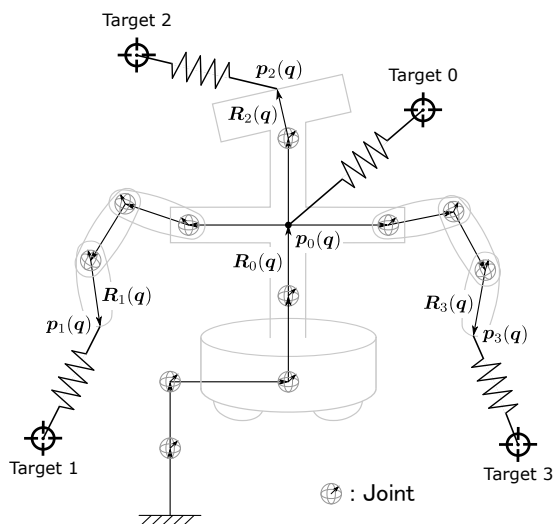


図5 仮想直進バネの概念図

る. 図5に仮想直進バネを示す(図中では仮想回転バネは省略されている). バネ定数を表す行列 \mathbf{K} を次のように定める.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_0 & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{K}_1 & \cdots & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{K}_r \end{bmatrix} \quad (17)$$

ここで, 対角行列 $\mathbf{K}_i \in \mathbb{R}^{6 \times 6}$ は誤差ベクトル $\mathbf{e}_i(\mathbf{q})$ に対して張られる仮想バネのバネ定数を表す. また, 仮想バネによって各関節に生じる仮想トルク $\boldsymbol{\tau}(\mathbf{q})$ は次式によって計算できる.

$$\boldsymbol{\tau}(\mathbf{q}) = \mathbf{J}^T(\mathbf{q}) \mathbf{K} \mathbf{e}(\mathbf{q}) \quad (18)$$

上式の $\mathbf{J}(\mathbf{q})$ は第3章で定式化した速度比行列である. 仮想トルク $\boldsymbol{\tau}(\mathbf{q})$ に対し, 関節ベースでのダンピング効果により, 各関節に速度 $\dot{\mathbf{q}}$ が発生すると見なせば, $\dot{\mathbf{q}}$ は次のように表される.

$$\dot{\mathbf{q}} = \mathbf{D}^{-1}(\mathbf{q}) \boldsymbol{\tau}(\mathbf{q}) \quad (19)$$

ここで, $\mathbf{D}(\mathbf{q})$ は関節ダンピング係数行列である. $\mathbf{D}(\mathbf{q})$ の具体的な定め方は後述する.

4.3 逆運動学の数値解法への展開

反復計算により, 逆運動学の数値解を求めることを考える. 反復計算 k 回目における各関節変位を \mathbf{q}_k とし, 以降では, $\mathbf{e}(\mathbf{q}_k)$, $\boldsymbol{\tau}(\mathbf{q}_k)$, $\mathbf{D}(\mathbf{q}_k)$, $\mathbf{J}(\mathbf{q}_k)$ をそれぞれ \mathbf{e}_k , $\boldsymbol{\tau}_k$, \mathbf{D}_k , \mathbf{J}_k と略記する.

仮想バネと関節ダンピング効果により各関節に速度が生じたとき, その速度によって各関節変位が1秒間当たりに変化する量を $\Delta \mathbf{q}_k$ とする. このとき, $\Delta \mathbf{q}_k$ は次のように表される.

$$\Delta \mathbf{q}_k = \mathbf{D}_k^{-1} \boldsymbol{\tau}_k = \mathbf{D}_k^{-1} \mathbf{J}_k^T \mathbf{K} \mathbf{e}_k \quad (20)$$

各関節変位 \mathbf{q}_k に $\Delta \mathbf{q}_k$ を加えたものを次のステップにおける各関節変位とすれば, 関節変位の更新則は次のように表される.

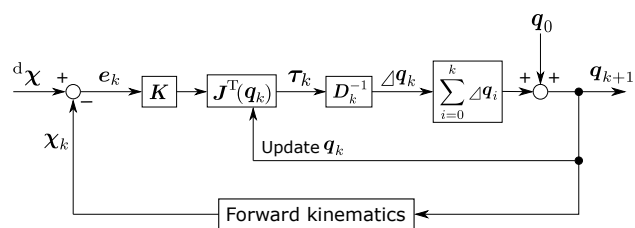


図6 逆運動学の計算手順を表すブロック線図

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta \mathbf{q}_k \quad (21)$$

また, \mathbf{q}_{k+1} は次のように書くこともできる.

$$\mathbf{q}_{k+1} = \mathbf{q}_0 + \sum_{i=0}^k \Delta \mathbf{q}_i \quad (22)$$

\mathbf{q}_0 は初期関節変位である. 上記の反復計算をブロック線図にまとめると図6のようになる. 図6では, 集合 ${}^d\boldsymbol{\chi}$, $\boldsymbol{\chi}_k$ をそれぞれ次のように定めた.

$${}^d\boldsymbol{\chi} = \{ {}^d\mathbf{p}_0, {}^d\mathbf{R}_0, \dots, {}^d\mathbf{p}_r, {}^d\mathbf{R}_r \} \quad (23)$$

$$\boldsymbol{\chi}_k = \{ \mathbf{p}_0(\mathbf{q}_k), \mathbf{R}_0(\mathbf{q}_k), \dots, \mathbf{p}_r(\mathbf{q}_k), \mathbf{R}_r(\mathbf{q}_k) \} \quad (24)$$

4.4 仮想バネの弾性エネルギー

仮想バネの弾性エネルギー V_k は以下のように表される.

$$V_k = \frac{1}{2} \mathbf{e}_k^T \mathbf{K} \mathbf{e}_k \in \mathbb{R} \quad (25)$$

V_k が最小となるような各関節変位が逆運動学解となる. よって, $\Delta \mathbf{q}_k$ に含まれる行列 \mathbf{D}_k は V_k の減少性を考慮して選定する必要がある. V_k を減少させるための \mathbf{D}_k の選定例として, 以下のものが提案されている [20].

$$\mathbf{D}_k = \mathbf{J}_k^T \mathbf{K} \mathbf{J}_k + \left(\frac{1}{2} V_k + \delta \right) \mathbf{I}_n \quad (26)$$

ここで, $\delta \in \mathbb{R}$ は $V_k \simeq 0$ のときに行列 \mathbf{D}_k の非正則化を防ぐための微小な正の定数である. 行列 $\mathbf{J}_k^T \mathbf{K} \mathbf{J}_k$ はインピーダンス制御 [22] やコンプライアンス制御 [23], 逆運動学の数値解法 [24] などで頻りに現れる. 行列 $\mathbf{J}_k^T \mathbf{K} \mathbf{J}_k$ のブロック分割については付録Bを参照されたい. 式 (26) に示した行列 \mathbf{D}_k は正定値対称行列となるため, その逆行列を計算する際は, Cholesky 分解 [25]~[28] を用いることで計算量を半減することができる. また, 行列 \mathbf{D}_k は $V_{k+1} \leq V_k$ となるための近似的な十分条件 [19] を満足している.

以上のように, 第2章と第3章で提案した運動学の定式化手法を用いることで, 分岐点や直進対偶を含む多関節ロボットにおいても, 逆運動学の数値解法を一つの理論にまとめることができるようになる.

5. シミュレーション実験

本章では, 第4章で述べた逆運動学の数値解法を双腕型の移動マニピュレータに適用し, シミュレーション実験を行う. その結果から, 提案手法のより具体的な応用例や各計算式の妥当性を示す.

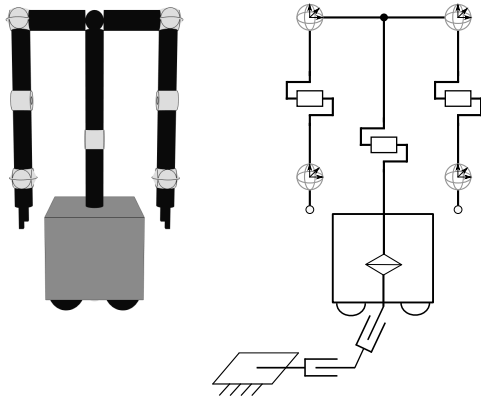


図7 シミュレーション実験で用いる多関節ロボットの運動学モデル（双腕型の移動マニピュレータ）

5.1 実験に使用した運動学モデルとパラメータ

シミュレーション実験に使用する多関節ロボットの運動学モデルを図7に示す。幹の自由度は4，枝（片腕）の自由度は7であり，全体の自由度は18である。片腕は2つの球対偶と1つの回転対偶によって構成されており，移動台車の変位は仮想的な2つの直進対偶と1つの回転対偶によって表現されている。ロボットの高さは1.4[m]，肩幅は0.8[m]である。また，ハンドの姿勢を表す回転行列は以下のようなになる。

$${}^hR_1 = {}^hR_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (27)$$

ロボットが基準姿勢 ($\mathbf{q} = \mathbf{0}$) のとき，リンクの姿勢を表す回転行列は全て単位行列であるのに対し，図7のロボットではハンドが真下を向いている。これを補正するための回転行列が hR_1 と hR_2 である。

本実験で使用したパネ定数行列 \mathbf{K} と定数 δ の値を以下に示す。

$$\mathbf{K} = \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{K}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{K}_2 \end{bmatrix}, \quad \delta = 0.02 \quad (28)$$

$$\mathbf{K}_1 = \mathbf{K}_2 = \text{diag}(1, 1, 1, 4.13, 4.13, 4.13) \quad (29)$$

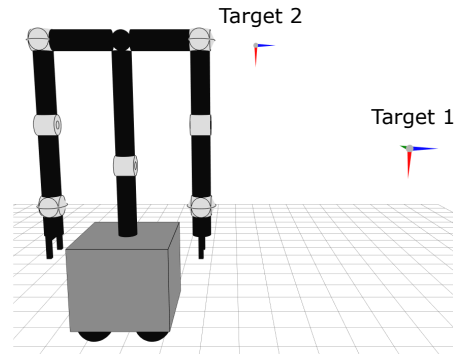
本実験では，幹の終点（分岐点）に対しては目標の位置・姿勢を与えないため， $\mathbf{K}_0 = \mathbf{O}$ とした。

5.2 逆運動学の数値計算例

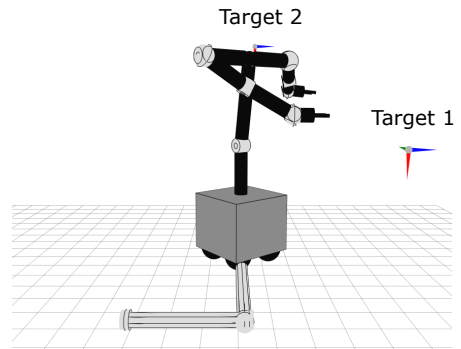
目標の位置・姿勢は，右腕の手先と左腕の手先に対して与えることにする。目標の位置・姿勢はそれぞれ次のように定めた。

$${}^d\mathbf{p}_1 = \begin{bmatrix} 1.8 \\ 0.8 \\ 0.8 \end{bmatrix} [\text{m}], \quad {}^d\mathbf{R}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (30)$$

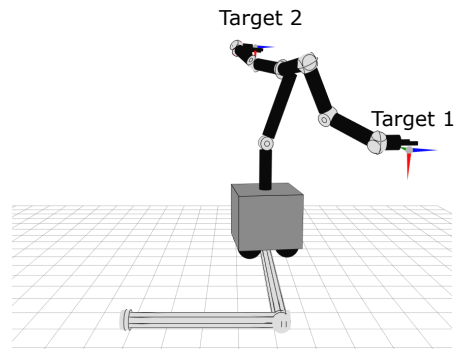
$${}^d\mathbf{p}_2 = \begin{bmatrix} 0.8 \\ 2.4 \\ 1.6 \end{bmatrix} [\text{m}], \quad {}^d\mathbf{R}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (31)$$



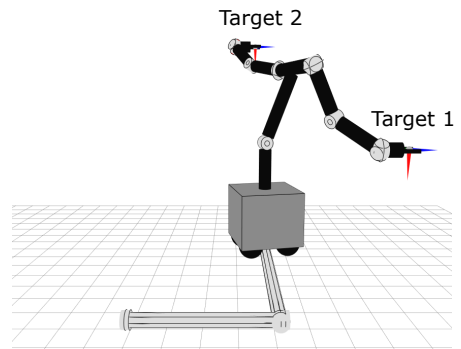
(a) 初期姿勢 ($k = 0$)



(b) 反復計算4回目 ($k = 4$)



(c) 反復計算8回目 ($k = 8$)



(d) 反復計算15回目 ($k = 15$)

図8 逆運動学の数値解法を実行したときに手先が目標の位置・姿勢に収束するまでの過程

手先が目標の位置・姿勢に収束するまでの過程を図8に示す。仮想バネの弾性エネルギー V_k がゼロ (0.001 以下) となるまでに 15 回の反復回数を要した。

移動台車の位置・姿勢は q_{00} , q_{01} , q_{02} から得られるため、逆運動学を解くことで移動台車の位置決め問題も同時に解くことができる。ただし、逆運動学の計算結果から得られるのは、移動台車が最終的に到達する位置・姿勢であり、そこに至るまでの経路を計算するためには、逆運動学とは別に移動台車の経路計画を行う必要がある。

以上より、提案した順運動学と速度比行列 (ヤコビ行列) の定式化手法を導入した場合においても、シミュレーション実験で逆運動学の数値解を計算できることが示された。すなわち、本論文で提案した各計算式の妥当性がシミュレーション実験の結果からも示された。

なお、2つの目標が離れた位置に与えられた場合、ロボットはどちらか一方の目標にしか到達することができない。このような状況を想定して、それぞれの目標に優先度を付与する方法も提案されている [29]~[31]。これらは優先度付き逆運動学と呼ばれており、例えば文献 [31] に示されている方法では、バネ定数行列 K に変数を1つ追加するだけで、それぞれの目標に優先度を付与することができる。

6. 結 言

本論文では、ロボットの構造が多様化した近年においても、汎用的に実用化可能な運動学を確立するため、ロボット運動学の基本を見直し、再定義を行った。提案手法を用いることで、分岐点や直進対偶、移動能力を有するロボットにおいても、同一の手順で運動学を計算することが可能となる。第5章の数値計算例では、双腕型の移動マニピュレータを例に用いたが、同様な手順で四足歩行ロボットや人型ロボットなどの逆運動学を計算することも可能である。

本論文ではロボットに含まれる分岐点の数を1つに限定して定式化を行ったため、分岐点を2つ以上有するロボットに対しては、提案手法を適用することができない。分岐点の数を一般化した形式に拡張し、汎用性をより高めることが本研究における今後の課題となっている。

参 考 文 献

- [1] R.S. Hartenberg and J. Denavit: "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", *Journal of Mechatronics*, vol.77, no.2, pp.215-221, 1955.
- [2] J.M. Renders, E. Rossignol, M. Becquet and R. Hanus: "Kinematic Calibration and Geometrical Parameter Identification for Robots", *IEEE Transactions on Robotics and Automation*, vol.7, no.6, pp.721-732, 1991.
- [3] W. Veitschegger and Chi-Haur Wu: "Robot Accuracy Analysis Based on Kinematics", *IEEE Journal on Robotics and Automation*, vol.2, no.3, pp.171-179, 1986.
- [4] S. Hayati and M. Mirmirani: "Improving the Absolute Positioning Accuracy of Robot Manipulators", *Journal of Robotic Systems*, vol.2, no.4, pp.397-413, 1985.
- [5] S. Hayati, K. Tso and G. Roston: "Robot Geometry Calibration", *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, vol.2, pp.947-951, 1988.
- [6] H. Zhuang and Z.S. Roth: "A Linear Solution to the Kinematic Parameter Identification of Robot Manipulators", *IEEE Transactions on Robotics and Automation*, vol.9, no.2, pp.174-185, 1993.
- [7] I.M. Chen, G. Yang, C.T. Tan and S.H. Yeo: "Local POE Model for Robot Kinematic Calibration", *Mechanism and Machine Theory*, vol.36, no.11-12, pp.1215-1239, 2001.
- [8] F. C. Park and K. Okamura: "Kinematic Calibration and the Product of Exponentials Formula", *Advances in Robot Kinematics and Computational Geometry*, MIT press, pp.119-128, 1994.
- [9] C. Wu: "The Kinematic Error Model for the Design of Robot Manipulator", *1983 American Control Conference*, pp.497-502, 1983.
- [10] P. I. Corke: "A Simple and Systematic Approach to Assigning Denavit-Hartenberg Parameters", *IEEE Transactions on Robotics*, vol.23, no.3, pp.590-594, 2007.
- [11] 大隅久: "ロボット工学の基礎", *精密工学会誌*, vol.73, no.10, pp.1123-1126, 2007.
- [12] A. Codourey and E. Burdet: "A Body-oriented Method for Finding a Linear Form of the Dynamic Equation of Fully Parallel Robots", *Proceedings of International Conference on Robotics and Automation*, vol.2, pp.1612-1618, 1997.
- [13] O. Khatib: "Inertial Properties in Robotic Manipulation: An Object-Level Framework", *The International Journal of Robotics Research*, vol.14, no.1, pp.19-36, 1995.
- [14] B. Siciliano: "The Tricept robot: Inverse Kinematics, Manipulability Analysis and Closed-Loop Direct Kinematics Algorithm", *Robotica*, vol.17, no.4, pp.437-445, 1999.
- [15] G. Tevatia and S. Schaal: "Inverse Kinematics for Humanoid Robots", *IEEE International Conference on Robotics and Automation*, vol.1, pp.294-299, 2000.
- [16] R. Campa, K. Camarillo and L. Arias: "Kinematic Modeling and Control of Robot Manipulators via Unit Quaternions: Application to a Spherical Wrist", *Proceedings of the 45th IEEE Conference on Decision and Control*, pp.6474-6479, 2006.
- [17] T. Sugihara and M. Morisawa: "A Survey: Dynamics of Humanoid Robots", *Advanced Robotics*, vol.34, no.21-22, pp.1338-1352, 2020.
- [18] 杉原知道: "逆運動学の数値解法", *日本ロボット学会誌*, vol.34, no.3, pp.167-173, 2016.
- [19] 関口敏範, 武居直行: "仮想バネの弾性エネルギー最小化に基づいた特異姿勢・非可解な問題に対して安定な逆運動学の数値解法", *日本ロボット学会誌*, vol.36, no.9, pp.645-653, 2018.
- [20] T. Sugihara: "Solvability-unconcerned Inverse Kinematics by Levenberg-Marquardt Method", *IEEE Transaction on Robotics*, vol.27, no.5, pp.984-991, 2011.
- [21] 関口敏範, 武居直行: "特異点における角軸ベクトルの計算法と拡張角軸ベクトルの提案", *日本ロボット学会誌*, vol.37, no.8, pp.726-734, 2019.
- [22] C. Ott, A. Albu-Schaffer, A. Kugi and G. Hirzinger: "On the Passivity-Based Impedance Control of Flexible Joint Robots", *IEEE Transactions on Robotics*, vol.24, no.2, pp.416-429, 2008.
- [23] J. K. Salisbury: "Active Stiffness Control of a Manipulator in Cartesian Coordinates", *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1980.
- [24] A. S. Deo and I. D. Walker: "Overview of Damped Least-Squares Methods for Inverse Kinematics of Robot Manipulators", *Journal of Intelligent and Robotic Systems*, vol.14, pp.43-68, 1995.
- [25] A. Krishnamoorthy and D. Menon: "Matrix inversion using Cholesky decomposition", *2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, pp.70-72, 2013.
- [26] A. W. Bojanczyk, R. P. Brent, P. van Dooren, and F. R. de Hoog: "A Note on DOWNDATING the Cholesky Factorization", *SIAM Journal on Scientific and Statistical Computing*, vol.8, no.3, pp.210-221, 1987.
- [27] N. J. Higham: "Cholesky factorization", *WIREs Computational Statistics*, vol.1, no.2, pp.251-254, 2009.
- [28] J. Li and O. B. Widlund: "FETI-DP, BDDC, and Block Cholesky Methods", *International Journal for Numerical Methods in Engineering*, vol.66, no.2, pp.250-271, 2006.
- [29] T. Sugihara: "Robust Solution of Prioritized Inverse Kinematics Based on Hestenes-Powell Multiplier Method", *IEEE/RSJ*

International Conference on Intelligent Robots and Systems, pp.510-515, 2014.

- [30] 関口叡範, 武居直行: “切断法による優先度付き逆運動学の数値解法”, 日本ロボット学会誌, vol.37, no.8, pp.711-717, 2019.
- [31] M. Sekiguchi and N. Takesue: “Fast and Robust Numerical Method for Inverse Kinematics with Prioritized Multiple Targets for Redundant Robots”, Advanced Robotics, vol.34, no.16, pp.1068-1078, 2020.

付録 A. 付録 : Rodrigues の回転公式

回転角度 q と単位回転軸ベクトル $\phi = [\phi_x \ \phi_y \ \phi_z]^T$ による回転変換を回転行列で表すと以下のようなになる。

$$\mathbf{R}(q, \phi) = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \in SO(3) \quad (\text{A.1})$$

$$\mathbf{r}_1 = \begin{bmatrix} \cos q + \phi_x^2(1 - \cos q) \\ \phi_x\phi_y(1 - \cos q) + \phi_z \sin q \\ \phi_z\phi_x(1 - \cos q) - \phi_y \sin q \end{bmatrix} \in \mathbb{R}^3 \quad (\text{A.2})$$

$$\mathbf{r}_2 = \begin{bmatrix} \phi_x\phi_y(1 - \cos q) - \phi_z \sin q \\ \cos q + \phi_y^2(1 - \cos q) \\ \phi_y\phi_z(1 - \cos q) + \phi_x \sin q \end{bmatrix} \in \mathbb{R}^3 \quad (\text{A.3})$$

$$\mathbf{r}_3 = \begin{bmatrix} \phi_z\phi_x(1 - \cos q) + \phi_y \sin q \\ \phi_y\phi_z(1 - \cos q) - \phi_x \sin q \\ \cos q + \phi_z^2(1 - \cos q) \end{bmatrix} \in \mathbb{R}^3 \quad (\text{A.4})$$

上記の式は Rodrigues の回転公式と呼ばれている。

付録 B. 行列 $\mathbf{J}^T \mathbf{K} \mathbf{J}$ のブロック分割

行列 $\mathbf{J}^T \mathbf{K} \mathbf{J}$ は二重縁付きブロック対角行列であり, 次のようにブロック分割することができる。

$$\mathbf{J}^T \mathbf{K} \mathbf{J} = \begin{bmatrix} \mathbf{C}_{00} & \mathbf{C}_{10}^T & \mathbf{C}_{20}^T & \cdots & \mathbf{C}_{r0}^T \\ \mathbf{C}_{10} & \mathbf{C}_{11} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{C}_{20} & \mathbf{O} & \mathbf{C}_{22} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{r0} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{C}_{rr} \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (\text{B.5})$$

ここで,

$$\mathbf{C}_{00} = \sum_{i=0}^r {}^s \mathbf{J}_i^T \mathbf{K}_i {}^s \mathbf{J}_i \in \mathbb{R}^{n_0 \times n_0} \quad (\text{B.6})$$

$$\mathbf{C}_{ii} = {}^b \mathbf{J}_i^T \mathbf{K}_i {}^b \mathbf{J}_i \in \mathbb{R}^{n_i \times n_i} \quad (i = 1, 2, \dots, r) \quad (\text{B.7})$$

$$\mathbf{C}_{i0} = {}^b \mathbf{J}_i^T \mathbf{K}_i {}^s \mathbf{J}_i \in \mathbb{R}^{n_i \times n_0} \quad (i = 1, 2, \dots, r) \quad (\text{B.8})$$

である。行列 $\mathbf{J}^T \mathbf{K} \mathbf{J}$ は半正定値行列であり, 正則とは限らない。例えば, 行列 \mathbf{J} が正方行列でないならば, 行列 $\mathbf{J}^T \mathbf{K} \mathbf{J}$ は正則にはならない。よって, 逆行列を計算する際は, 減衰因子を追加するなどの正則化が必要である。