

軌道最適化による動作生成 リファレンスマニュアル

平成 31 年 2 月 8 日

室岡雅樹

murooka@jsk.t.u-tokyo.ac.jp

目 次

| | | |
|-------|----------------------------------|----|
| 1 | 軌道最適化による動作生成の基礎 | 1 |
| 1.1 | タスク関数のノルムを最小にするコンフィギュレーションの探索 | 1 |
| 1.2 | コンフィギュレーション二次形式の正則化項の追加 | 2 |
| 1.3 | コンフィギュレーション更新量の正則項の追加 | 3 |
| 1.4 | ソースコードと数式の対応 | 3 |
| 1.5 | 章の構成 | 4 |
| 2 | コンフィギュレーションとタスク関数 | 4 |
| 2.1 | 瞬時コンフィギュレーションと瞬時タスク関数 | 4 |
| 2.2 | 軌道コンフィギュレーションと軌道タスク関数 | 21 |
| 3 | 逐次二次計画法 | 28 |
| 4 | 動作生成の拡張 | 30 |
| 4.1 | マニピュレーションの動作生成 | 30 |
| 4.2 | B スプラインを用いた関節軌道生成 | 41 |
| 4.2.1 | B スプラインを用いた関節軌道生成の基礎 | 41 |
| 4.2.2 | B スプラインを用いた関節軌道生成の実装 | 50 |
| 5 | 補足 | 62 |
| 5.1 | 既存のロボット基礎クラスの拡張 | 62 |
| 5.2 | 環境と接触するロボットの関節・リンク構造 | 65 |
| 5.3 | irteus の inverse-kinematics 互換関数 | 67 |
| 5.4 | 関節トルク勾配の計算 | 71 |

1 軌道最適化による動作生成の基礎

1.1 タスク関数のノルムを最小にするコンフィギュレーションの探索

$q \in \mathbb{R}^{n_q}$ を設計対象のコンフィギュレーションとする．例えば一般の逆運動学計算では， q はある瞬間のロボットの関節角度を表すベクトルで，コンフィギュレーションの次元 n_q はロボットの関節自由度数となる．

動作生成問題を，所望のタスクに対応するタスク関数 $e(q) : \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_e}$ について，次式を満たす q を得ることとして定義する．

$$e(q) = 0 \quad (1)$$

例えば一般の逆運動学計算では， $e(q)$ はエンドエフェクタの目標位置姿勢と現在位置姿勢の差を表す 6 次元ベクトルである．非線形方程式 (1) の解を解析的に得ることは難しく，反復計算による数値解法が採られる．式 (1) が解をもたないときでも最善のコンフィギュレーションを得られるように一般化すると，式 (1) の求解は次の最適化問題として表される¹．

$$\min_q F(q) \quad (2a)$$

$$\text{where } F(q) \stackrel{\text{def}}{=} \frac{1}{2} \|e(q)\|^2 \quad (2b)$$

コンフィギュレーションが最小値 q_{min} と最大値 q_{max} の間に含まれる必要があるとき，逆運動学計算は次の制約付き非線形最適化問題として表される．

$$\min_q F(q) \quad \text{s.t. } q_{min} \leq q \leq q_{max} \quad (3)$$

例えば一般の逆運動学計算では， q_{min}, q_{max} は関節角度の許容範囲の最小値，最大値を表す．以降では，式 (3) の制約を，より一般の形式である線形等式制約，線形不等式制約として次式のように表す²．

$$\min_q F(q) \quad (4a)$$

$$\text{s.t. } Aq = \bar{b} \quad (4b)$$

$$Cq \geq \bar{d} \quad (4c)$$

制約付き非線形最適化問題の解法のひとつである逐次二次計画法では，次の二次計画問題の最適解として得られる Δq_k^* を用いて， $q_{k+1} = q_k + \Delta q_k^*$ として反復更新することで，式 (4) の最適解を導出する³．

$$\min_{\Delta q_k} F(q_k) + \nabla F(q_k)^T \Delta q_k + \frac{1}{2} \Delta q_k^T \nabla^2 F(q_k) \Delta q_k \quad (5a)$$

$$\text{s.t. } A \Delta q_k = \bar{b} - Aq_k \quad (5b)$$

$$C \Delta q_k \geq \bar{d} - Cq_k \quad (5c)$$

¹ 任意の半正定値行列 W に対して， $\|e(q)\|_W^2 = e(q)^T W e(q) = e(q)^T S^T S e(q) = \|S e(q)\|^2$ を満たす S が必ず存在するので，式 (2b) は任意の重み付きノルムを表現可能である．

² 式 (3) における関節角度の最小値，最大値に関する制約は次式のように表される．

$$\begin{aligned} q_{min} &\leq q \leq q_{max} \\ \Leftrightarrow \begin{pmatrix} I \\ -I \end{pmatrix} q &\geq \begin{pmatrix} q_{min} \\ -q_{max} \end{pmatrix} \end{aligned}$$

³ 式 (5a) は $F(q)$ を q_k の周りでテーラー展開し三次以下の項を省略したものに一致する．逐次二次計画法については，以下の書籍の 18 章で詳しく説明されている．

Numerical optimization, S. Wright and J. Nocedal, Springer Science, vol. 35, 1999, http://www.xn--vjq503akpco3w.top/literature/Nocedal_Wright_Numerical_optimization_v2.pdf.

$\nabla F(\mathbf{q}_k), \nabla^2 F(\mathbf{q}_k)$ はそれぞれ, $F(\mathbf{q}_k)$ の勾配, ヘッセ行列⁴で, 次式で表される.

$$\nabla F(\mathbf{q}) = \left(\frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}} \right)^T \mathbf{e}(\mathbf{q}) \quad (6a)$$

$$= \mathbf{J}(\mathbf{q})^T \mathbf{e}(\mathbf{q}) \quad (6b)$$

$$\nabla^2 F(\mathbf{q}) = \sum_{i=1}^m e_i(\mathbf{q}) \nabla^2 e_i(\mathbf{q}) + \left(\frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}} \right)^T \frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}} \quad (6c)$$

$$\approx \left(\frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}} \right)^T \frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}} \quad (6d)$$

$$= \mathbf{J}(\mathbf{q})^T \mathbf{J}(\mathbf{q}) \quad (6e)$$

ただし, $e_i(\mathbf{q})$ ($i = 1, 2, \dots, m$) は $\mathbf{e}(\mathbf{q})$ の i 番目の要素である. 式 (6c) から式 (6d) への変形では $\mathbf{e}(\mathbf{q})$ の二階微分がゼロであると近似している. $\mathbf{J}(\mathbf{q}) \stackrel{\text{def}}{=} \frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{n_e \times n_q}$ は $\mathbf{e}(\mathbf{q})$ のヤコビ行列である.

式 (6a), 式 (6d) から式 (5a) の目的関数は次式で表される⁵.

$$\frac{1}{2} \mathbf{e}_k^T \mathbf{e}_k + \mathbf{e}_k^T \mathbf{J}_k \Delta \mathbf{q}_k + \frac{1}{2} \Delta \mathbf{q}_k^T \mathbf{J}_k^T \mathbf{J}_k \Delta \mathbf{q}_k \quad (7a)$$

$$= \frac{1}{2} \|\mathbf{e}_k + \mathbf{J}_k \Delta \mathbf{q}_k\|^2 \quad (7b)$$

ただし, $\mathbf{e}_k \stackrel{\text{def}}{=} \mathbf{e}(\mathbf{q}_k), \mathbf{J}_k \stackrel{\text{def}}{=} \mathbf{J}(\mathbf{q}_k)$ とした.

結局, 逐次二次計画法で反復的に解かれる二次計画問題 (5) は次式で表される.

$$\min_{\Delta \mathbf{q}_k} \frac{1}{2} \Delta \mathbf{q}_k^T \mathbf{J}_k^T \mathbf{J}_k \Delta \mathbf{q}_k + \mathbf{e}_k^T \mathbf{J}_k \Delta \mathbf{q}_k \quad (8a)$$

$$\text{s.t. } \mathbf{A} \Delta \mathbf{q}_k = \mathbf{b} \quad (8b)$$

$$\mathbf{C} \Delta \mathbf{q}_k \geq \mathbf{d} \quad (8c)$$

ここで,

$$\mathbf{b} = \bar{\mathbf{b}} - \mathbf{A} \mathbf{q}_k \quad (9)$$

$$\mathbf{d} = \bar{\mathbf{d}} - \mathbf{C} \mathbf{q}_k \quad (10)$$

とおいた.

1.2 コンフィギュレーション二次形式の正則化項の追加

式 (2a) の最適化問題の目的関数を, 次式の $\hat{F}(\mathbf{q})$ で置き換える.

$$\hat{F}(\mathbf{q}) = F(\mathbf{q}) + F_{reg}(\mathbf{q}) \quad (11)$$

$$\text{where } F_{reg}(\mathbf{q}) = \frac{1}{2} \mathbf{q}^T \bar{\mathbf{W}}_{reg} \mathbf{q} \quad (12)$$

目的関数 $\hat{F}(\mathbf{q})$ の勾配, ヘッセ行列は次式で表される.

$$\nabla \hat{F}(\mathbf{q}) = \nabla F(\mathbf{q}) + \nabla F_{reg}(\mathbf{q}) \quad (13a)$$

$$= \mathbf{J}(\mathbf{q})^T \mathbf{e}(\mathbf{q}) + \bar{\mathbf{W}}_{reg} \mathbf{q} \quad (13b)$$

$$\nabla^2 \hat{F}(\mathbf{q}) = \nabla^2 F(\mathbf{q}) + \nabla^2 F_{reg}(\mathbf{q}) \quad (13c)$$

$$\approx \mathbf{J}(\mathbf{q})^T \mathbf{J}(\mathbf{q}) + \bar{\mathbf{W}}_{reg} \quad (13d)$$

⁴式 (5a) の $\nabla^2 F(\mathbf{q}_k)$ の部分は一般にはラグランジュ関数の \mathbf{q}_k に関するヘッセ行列であるが, 等式・不等式制約が線形の場合は $F(\mathbf{q}_k)$ のヘッセ行列と等価になる.

⁵式 (7b) は, 以下の論文で紹介されている二次計画法によってコンフィギュレーション速度を導出する逆運動学解法における目的関数と一致する.

Feasible pattern generation method for humanoid robots, F. Kanehiro et al., Proceedings of the 2009 IEEE-RAS International Conference on Humanoid Robots, pp. 542-548, 2009.

したがって，式 (8) の二次計画問題は次式で表される．

$$\min_{\Delta \mathbf{q}_k} \frac{1}{2} \Delta \mathbf{q}_k^T \left(\mathbf{J}_k^T \mathbf{J}_k + \bar{\mathbf{W}}_{reg} \right) \Delta \mathbf{q}_k + \left(\mathbf{J}_k^T \mathbf{e}_k + \bar{\mathbf{W}}_{reg} \mathbf{q}_k \right)^T \Delta \mathbf{q}_k \quad (14a)$$

$$\text{s.t. } \mathbf{A} \Delta \mathbf{q}_k = \mathbf{b} \quad (14b)$$

$$\mathbf{C} \Delta \mathbf{q}_k \geq \mathbf{d} \quad (14c)$$

1.3 コンフィギュレーション更新量の正則項の追加

Gauss-Newton 法と Levenberg-Marquardt 法の比較を参考に，式 (14a) の二次形式項の行列に，次式のように微小な係数をかけた単位行列を加えると，一部の適用例について逐次二次計画法の収束性が改善された⁶．

$$\min_{\Delta \mathbf{q}_k} \frac{1}{2} \Delta \mathbf{q}_k^T \left(\mathbf{J}_k^T \mathbf{J}_k + \bar{\mathbf{W}}_{reg} + \lambda \mathbf{I} \right) \Delta \mathbf{q}_k + \left(\mathbf{J}_k^T \mathbf{e}_k + \bar{\mathbf{W}}_{reg} \mathbf{q}_k \right)^T \Delta \mathbf{q}_k \quad (15a)$$

$$\text{s.t. } \mathbf{A} \Delta \mathbf{q}_k = \mathbf{b} \quad (15b)$$

$$\mathbf{C} \Delta \mathbf{q}_k \geq \mathbf{d} \quad (15c)$$

改良誤差減衰最小二乗法⁷を参考にすると， λ は次式のように決定される．

$$\lambda = \lambda_r F(\mathbf{q}_k) + w_r \quad (16)$$

λ_r と w_r は正の定数である．

1.4 ソースコードと数式の対応

$$\mathbf{W}_{reg} \stackrel{\text{def}}{=} \bar{\mathbf{W}}_{reg} + \lambda \mathbf{I} \quad (17a)$$

$$\mathbf{v}_{reg} \stackrel{\text{def}}{=} \bar{\mathbf{W}}_{reg} \mathbf{q}_k \quad (17b)$$

とすると，式 (15) は次式で表される．

$$\min_{\Delta \mathbf{q}_k} \frac{1}{2} \Delta \mathbf{q}_k^T \left(\mathbf{J}_k^T \mathbf{J}_k + \mathbf{W} \right) \Delta \mathbf{q}_k + \left(\mathbf{J}_k^T \mathbf{e}_k + \mathbf{v}_{reg} \right)^T \Delta \mathbf{q}_k \quad (18a)$$

$$\text{s.t. } \mathbf{A} \Delta \mathbf{q}_k = \mathbf{b} \quad (18b)$$

$$\mathbf{C} \Delta \mathbf{q}_k \geq \mathbf{d} \quad (18c)$$

第 2 節や第 4 章で説明する ****-configuration-task* クラスのメソッドは式 (18) 中の記号と以下のように対応している．

| | |
|---|---|
| <i>:config-vector</i> | get \mathbf{q} |
| <i>:set-config</i> | set \mathbf{q} |
| <i>:task-value</i> | get $\mathbf{e}(\mathbf{q})$ |
| <i>:task-jacobian</i> | get $\mathbf{J}(\mathbf{q}) \stackrel{\text{def}}{=} \frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}}$ |
| <i>:config-equality-constraint-matrix</i> | get \mathbf{A} |
| <i>:config-equality-constraint-vector</i> | get \mathbf{b} |
| <i>:config-inequality-constraint-matrix</i> | get \mathbf{C} |
| <i>:config-inequality-constraint-vector</i> | get \mathbf{d} |
| <i>:regular-matrix</i> | get \mathbf{W}_{reg} |
| <i>:regular-vector</i> | get \mathbf{v}_{reg} |

⁶これは，最適化における信頼領域 (trust region) に関連している．

⁷ Levenberg-Marquardt 法による可解性を問わない逆運動学，杉原 知道，日本ロボット学会誌，vol. 29，no. 3，pp. 269-277，2011.

1.5 章の構成

第2章では，コンフィギュレーション q の取得・更新，タスク関数 $e(q)$ の取得，タスク関数のヤコビ行列 $J(q) \stackrel{\text{def}}{=} \frac{\partial e(q)}{\partial q}$ の取得，コンフィギュレーションの等式・不等式制約 A, b, C, d の取得のためのクラスを説明する．第2.1節ではコンフィギュレーション q が瞬時の情報，第2.2節ではコンフィギュレーション q が時系列の情報を表す場合をそれぞれ説明する．

第3章では，第2章で説明されるクラスを用いて逐次二次計画法により最適化を行うためのクラスを説明する．

第4章では，用途に応じて拡張されたコンフィギュレーションとタスク関数のクラスを説明する．第4.1節では，マニピュレーションのために，ロボットに加えて物体のコンフィギュレーションを計画する場合を説明する．第4.2節では，ロボットの関節位置の軌道をBスプライン関数でパラメトリックに表現する場合を説明する．いずれにおいても，最適化では第3章で説明された逐次二次計画法のクラスが利用される．

第5章では，その他の補足事項を説明する．第5.1節では，jskeusで定義されているクラスの拡張について説明する．第5.2節では，環境との接触を有するロボットの問題設定を記述するためのクラスについて説明する．第5.4節では，関節トルクを関節角度で微分したヤコビ行列を導出するための関数について説明する．

2 コンフィギュレーションとタスク関数

2.1 瞬時コンフィギュレーションと瞬時タスク関数

instant-configuration-task

[class]

```

:super      propertied-object
:slots      (_robot-env robot-environment instance)
             (_theta-vector  $\theta$  [rad] [m])
             (_wrench-vector  $\hat{w}$  [N] [Nm])
             (_torque-vector  $\tau$  [Nm])
             (_phi-vector  $\phi$  [rad] [m])
             (_num-kin  $N_{kin} := |\mathcal{T}^{kin-trg}| = |\mathcal{T}^{kin-att}|$ )
             (_num-contact  $N_{cnt} := |\mathcal{T}^{cnt-trg}| = |\mathcal{T}^{cnt-att}|$ )
             (_num-variant-joint  $N_{var-joint} := |\mathcal{J}_{var}|$ )
             (_num-invariant-joint  $N_{invar-joint} := |\mathcal{J}_{invar}|$ )
             (_num-drive-joint  $N_{drive-joint} := |\mathcal{J}_{drive}|$ )
             (_num-posture-joint  $N_{posture-joint} := |\mathcal{J}_{posture}|$ )
             (_num-external  $N_{ex} :=$  number of external wrenches)
             (_num-collision  $N_{col} :=$  number of collision check pairs)
             (_dim-theta  $dim(\theta) = N_{var-joint}$ )
             (_dim-wrench  $dim(\hat{w}) = 6N_{cnt}$ )
             (_dim-torque  $dim(\tau) = N_{drive-joint}$ )
             (_dim-phi  $dim(\phi) = N_{invar-joint}$ )
             (_dim-variant-config  $dim(q_{var})$ )
             (_dim-invariant-config  $dim(q_{invar})$ )
             (_dim-config  $dim(q)$ )
             (_dim-task  $dim(e)$ )
             (_kin-scale-mat-list  $K_{kin}$ )
             (_target-posture-scale  $k_{posture}$ )

```

(_norm-regular-scale-max k_{max})
 (_norm-regular-scale-offset k_{off})
 (_torque-regular-scale k_{trq})
 (_variant-joint-list \mathcal{J}_{var})
 (_invariant-joint-list \mathcal{J}_{invar})
 (_drive-joint-list \mathcal{J}_{drive})
 (_kin-target-coords-list $\mathcal{T}^{kin-trg}$)
 (_kin-attention-coords-list $\mathcal{T}^{kin-att}$)
 (_contact-target-coords-list $\mathcal{T}^{cnt-trg}$)
 (_contact-attention-coords-list $\mathcal{T}^{cnt-att}$)
 (_variant-joint-angle-margin margin of θ [deg] [mm])
 (_invariant-joint-angle-margin margin of ϕ [deg] [mm])
 (_delta-linear-joint trust region of linear joint configuration [mm])
 (_delta-rotational-joint trust region of rotational joint configuration [deg])
 (_contact-constraint-list list of contact-constraint instance)
 (_posture-joint-list $\mathcal{J}_{posture}$)
 (_posture-joint-angle-list $\bar{\theta}^{trg}$)
 (_external-wrench-list $\{\mathbf{w}_1^{ex}, \mathbf{w}_2^{ex}, \dots, \mathbf{w}_{N_{ex}}^{ex}\}$)
 (_external-coords-list $\{T_1^{ex}, T_2^{ex}, \dots, T_{N_{ex}}^{ex}\}$)
 (_collision-pair-list list of bodyset-link or body pair)
 (_collision-distance-margin-list list of collision distance margin)
 (_only-kinematics? whether to consider only kinematics or not)
 (_variant-task-jacobi buffer for $\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}}$)
 (_invariant-task-jacobi buffer for $\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}}$)
 (_task-jacobi buffer for $\frac{\partial \mathbf{e}}{\partial \mathbf{q}}$)
 (_collision-theta-inequality-constraint-matrix buffer for $\mathbf{C}_{col, \theta}$)
 (_collision-phi-inequality-constraint-matrix buffer for $\mathbf{C}_{col, \phi}$)
 (_collision-inequality-constraint-vector buffer for \mathbf{C}_{col})

瞬時コンフィギュレーション $\mathbf{q}^{(l)}$ と瞬時タスク関数 $e^{(l)}(\mathbf{q}^{(l)})$ のクラス .

このクラスの説明で用いる全ての変数は、時間ステップ l を表す添字をつけて $x^{(l)}$ と表されるべきだが、このクラス内の説明では省略して x と表す . また、以降では、説明文やメソッド名で、“瞬時” や “instant” を省略する .

コンフィギュレーション \mathbf{q} の取得・更新、タスク関数 $e(\mathbf{q})$ の取得、タスク関数のヤコビ行列 $\frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}}$ の取得、コンフィギュレーションの等式・不等式制約 A, b, C, d の取得のためのメソッドが定義されている . コンフィギュレーション・タスク関数を定めるために、初期化時に以下を与える

- ロボット・環境

robot-environment ロボット・環境を表す robot-environment クラスのインスタンス

variant-joint-list \mathcal{J}_{var} 時変関節

invariant-joint-list \mathcal{J}_{invar} 時不変関節 (与えなければ時不変関節は考慮されない)

drive-joint-list \mathcal{J}_{drive} 駆動関節 (与えなければ関節駆動トルクは考慮されない)

- 幾何拘束

kin-target-coords-list $\mathcal{T}^{kin-trg}$ 幾何到達目標位置姿勢リスト

kin-attention-coords-list $\mathcal{T}^{kin-att}$ 幾何到達着目位置姿勢リスト
 kin-scale-mat-list K_{kin} 幾何拘束の座標系, 重みを表す変換行列のリスト

- 接触拘束

contact-target-coords-list $\mathcal{T}^{cnt-trg}$ 接触目標位置姿勢リスト
 contact-attention-coords-list $\mathcal{T}^{cnt-att}$ 接触着目位置姿勢リスト
 contact-constraint-list 接触レンチ制約リスト

- コンフィギュレーション拘束 (必要な場合のみ)

posture-joint-list $\mathcal{J}_{posture}$ 着目関節リスト
 posture-joint-angle-list $\bar{\theta}^{trg}$ 着目関節の目標関節角
 target-posture-scale $k_{posture}$ コンフィギュレーション拘束の重み

- 干渉回避拘束 (必要な場合のみ)

collision-pair-list 干渉回避する bodyset-link もしくは body のペアのリスト
 collision-distance-margin 干渉回避の距離マージン (全てのペアで同じ値の場合)
 collision-distance-margin-list 干渉回避の距離マージンのリスト (ペアごとに異なる値の場合)

- 外レンチ (必要な場合のみ)

external-wrench-list 外レンチのリスト (ワールド座標系で表す)
 external-coords-list 外レンチの作用点座標のリスト (位置のみを使用)

- 目的関数の重み

norm-regular-scale-max k_{max} コンフィギュレーション更新量正則化の重み最大値
 norm-regular-scale-offset k_{off} コンフィギュレーション更新量正則化の重みオフセット
 torque-regular-scale k_{trq} トルク正則化の重み

コンフィギュレーション q は以下から構成される .

$$q := \begin{pmatrix} \theta^T & \hat{w}^T & \tau^T & \phi^T \end{pmatrix}^T \quad (19)$$

$\theta \in \mathbb{R}^{N_{var-joint}}$ 時変関節角度 [rad] [m]

$\hat{w} \in \mathbb{R}^{6N_{cnt}}$ 接触レンチ [N] [Nm]

$\tau \in \mathbb{R}^{N_{drive-joint}}$ 関節駆動トルク [Nm] [N]

$\phi \in \mathbb{R}^{N_{invar-joint}}$ 時不変関節角度 [rad] [m]

\hat{w} は次式のように, 全接触部位でのワールド座標系での力・モーメントを並べたベクトルである .

$$\hat{w} = \begin{pmatrix} w_1^T & w_2^T & \cdots & w_{N_{cnt}}^T \end{pmatrix}^T \quad (20)$$

$$= \begin{pmatrix} f_1^T & n_1^T & f_2^T & n_2^T & \cdots & f_{N_{cnt}}^T & n_{N_{cnt}}^T \end{pmatrix}^T \quad (21)$$

タスク関数 $e(q)$ は以下から構成される .

$$e(q) := \begin{pmatrix} e^{kinT}(q) & e^{eom-transT}(q) & e^{eom-rotT}(q) & e^{trqT}(q) & e^{postureT}(q) \end{pmatrix}^T \quad (22)$$

$e^{kin}(q) \in \mathbb{R}^{6N_{kin}}$ 幾何到達拘束 [rad] [m]

$e^{eom-trans}(q) \in \mathbb{R}^3$ 力の釣り合い [N]

$e^{eom-rot}(q) \in \mathbb{R}^3$ モーメントの釣り合い [Nm]

$e^{trq}(q) \in \mathbb{R}^{N_{drive-joint}}$ 関節トルクの釣り合い [rad] [m]

$e^{posture}(\mathbf{q}) \in \mathbb{R}^{N_{posture-joint}}$ 関節角目標 [rad] [m]

| | | |
|------------------------------|---|----------|
| :init | <i>ℰkey (name)</i> <i>(robot-env)</i> <i>(variant-joint-list (send robot-env :variant-joint-list))</i> <i>(invariant-joint-list (send robot-env :invariant-joint-list))</i> <i>(drive-joint-list (send robot-env :drive-joint-list))</i> <i>(only-kinematics?)</i> <i>(kin-target-coords-list)</i> <i>(kin-attention-coords-list)</i> <i>(contact-target-coords-list)</i> <i>(contact-attention-coords-list)</i> <i>(variant-joint-angle-margin 3.0)</i> <i>(invariant-joint-angle-margin 3.0)</i> <i>(delta-linear-joint)</i> <i>(delta-rotational-joint)</i> <i>(contact-constraint-list (send-all contact-attention-coords-list :get :contact-constraint))</i> <i>(posture-joint-list)</i> <i>(posture-joint-angle-list)</i> <i>(external-wrench-list)</i> <i>(external-coords-list)</i> <i>(collision-pair-list)</i> <i>(collision-distance-margin 0.01)</i> <i>(collision-distance-margin-list)</i> <i>(kin-scale 1.0)</i> <i>(kin-scale-list)</i> <i>(kin-scale-mat-list)</i> <i>(target-posture-scale 0.001)</i> <i>(norm-regular-scale-max (if only-kinematics? 0.001 1.000000e-05))</i> <i>(norm-regular-scale-offset 1.000000e-07)</i> <i>(torque-regular-scale 1.000000e-04)</i> <i>ℰallow-other-keys</i> | [method] |
| | Initialize instance | |
| :robot-env | return robot-environment instance | [method] |
| :variant-joint-list | return \mathcal{J}_{var} | [method] |
| :invariant-joint-list | return \mathcal{J}_{invar} | [method] |
| :drive-joint-list | return \mathcal{J}_{drive} | [method] |
| :only-kinematics? | return whether to consider only kinematics or not | [method] |

| | |
|---|---|
| :theta | [method] |
| return $\boldsymbol{\theta}$ | |
| :wrench | [method] |
| return $\hat{\boldsymbol{w}}$ | |
| :torque | [method] |
| return $\boldsymbol{\tau}$ | |
| :phi | [method] |
| return $\boldsymbol{\phi}$ | |
| :num-kin | [method] |
| return $N_{kin} := \mathcal{T}^{kin-trg} = \mathcal{T}^{kin-att} $ | |
| :num-contact | [method] |
| return $N_{cnt} := \mathcal{T}^{cnt-trg} = \mathcal{T}^{cnt-att} $ | |
| :num-variant-joint | [method] |
| return $N_{var-joint} := \mathcal{J}_{var} $ | |
| :num-invariant-joint | [method] |
| return $N_{invar-joint} := \mathcal{J}_{invar} $ | |
| :num-drive-joint | [method] |
| return $N_{drive-joint} := \mathcal{J}_{drive} $ | |
| :num-posture-joint | [method] |
| return $N_{target-joint} := \mathcal{J}_{target} $ | |
| :num-external | [method] |
| return $N_{ex} :=$ number of external wrench | |
| :num-collision | [method] |
| return $N_{col} :=$ number of collision check pairs | |
| :dim-variant-config | [method] |
| | |
| | $dim(\boldsymbol{q}_{var}) \quad := \quad dim(\boldsymbol{\theta}) + dim(\hat{\boldsymbol{w}}) + dim(\boldsymbol{\tau}) \quad (23)$ |
| | $= \quad N_{var-joint} + 6N_{cnt} + N_{drive-joint} \quad (24)$ |
| return $dim(\boldsymbol{q}_{var})$ | |
| :dim-invariant-config | [method] |
| return $dim(\boldsymbol{q}_{invar}) := dim(\boldsymbol{\phi}) = N_{invar-joint}$ | |
| :dim-config | [method] |
| return $dim(\boldsymbol{q}) := dim(\boldsymbol{q}_{var}) + dim(\boldsymbol{q}_{invar})$ | |
| :dim-task | [method] |

$$dim(\boldsymbol{e}) \quad := \quad dim(\boldsymbol{e}^{kin}) + dim(\boldsymbol{e}^{com-trans}) + dim(\boldsymbol{e}^{com-rot}) + dim(\boldsymbol{e}^{trq}) + dim(\boldsymbol{e}^{posture}) \quad (25)$$

$$= \quad 6N_{kin} + 3 + 3 + N_{drive-joint} + N_{posture-joint} \quad (26)$$

return $\dim(\mathbf{e})$

:variant-config-vector

[method]

return $\mathbf{q}_{var} := \begin{pmatrix} \boldsymbol{\theta} \\ \hat{\mathbf{w}} \\ \boldsymbol{\tau} \end{pmatrix}$

:invariant-config-vector

[method]

return $\mathbf{q}_{invar} := \boldsymbol{\phi}$

:config-vector

[method]

return $\mathbf{q} := \begin{pmatrix} \mathbf{q}_{var} \\ \mathbf{q}_{invar} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\theta} \\ \hat{\mathbf{w}} \\ \boldsymbol{\tau} \\ \boldsymbol{\phi} \end{pmatrix}$

:set-theta *theta-new* *ℰkey* (*relative?* *nil*)

[method]

(*apply-to-robot?* *t*)

Set $\boldsymbol{\theta}$.

:set-wrench *wrench-new* *ℰkey* (*relative?* *nil*)

[method]

Set $\hat{\mathbf{w}}$.

:set-torque *torque-new* *ℰkey* (*relative?* *nil*)

[method]

Set $\boldsymbol{\tau}$.

:set-phi *phi-new* *ℰkey* (*relative?* *nil*)

[method]

(*apply-to-robot?* *t*)

Set $\boldsymbol{\phi}$.

:set-variant-config *variant-config-new* *ℰkey* (*relative?* *nil*)

[method]

(*apply-to-robot?* *t*)

Set \mathbf{q}_{var} .

:set-invariant-config *invariant-config-new* *ℰkey* (*relative?* *nil*)

[method]

(*apply-to-robot?* *t*)

Set \mathbf{q}_{invar} .

:set-config *config-new* *ℰkey* (*relative?* *nil*)

[method]

(*apply-to-robot?* *t*)

Set \mathbf{q} .

:kin-target-coords-list

[method]

$$T_m^{kin-trg} = \{\mathbf{p}_m^{kin-trg}, \mathbf{R}_m^{kin-trg}\} \quad (m = 1, 2, \dots, N_{kin}) \quad (27)$$

return $\mathcal{T}^{kin-trg} := \{T_1^{kin-trg}, T_2^{kin-trg}, \dots, T_{N_{kin}}^{kin-trg}\}$

:kin-attention-coords-list

[method]

$$T_m^{kin-att} = \{\mathbf{p}_m^{kin-att}, \mathbf{R}_m^{kin-att}\} \quad (m = 1, 2, \dots, N_{kin}) \quad (28)$$

return $\mathcal{T}^{kin-att} := \{T_1^{kin-att}, T_2^{kin-att}, \dots, T_{N_{kin}}^{kin-att}\}$

:contact-target-coords-list

[method]

$$T_m^{cnt-trg} = \{\mathbf{p}_m^{cnt-trg}, \mathbf{R}_m^{cnt-trg}\} \quad (m = 1, 2, \dots, N_{cnt}) \quad (29)$$

return $\mathcal{T}^{cnt-trg} := \{T_1^{cnt-trg}, T_2^{cnt-trg}, \dots, T_{N_{cnt}}^{cnt-trg}\}$

:contact-attention-coords-list

[method]

$$T_m^{cnt-att} = \{\mathbf{p}_m^{cnt-att}, \mathbf{R}_m^{cnt-att}\} \quad (m = 1, 2, \dots, N_{cnt}) \quad (30)$$

return $\mathcal{T}^{cnt-att} := \{T_1^{cnt-att}, T_2^{cnt-att}, \dots, T_{N_{cnt}}^{cnt-att}\}$

:contact-constraint-list

[method]

return list of contact-constraint instance

:wrench-list

[method]

return $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{N_{cnt}}\}$

:force-list

[method]

return $\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{N_{cnt}}\}$

:moment-list

[method]

return $\{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_{N_{cnt}}\}$

:external-wrench-list

[method]

return $\{\mathbf{w}_1^{ex}, \mathbf{w}_2^{ex}, \dots, \mathbf{w}_{N_{ex}}^{ex}\}$

:external-force-list

[method]

return $\{\mathbf{f}_1^{ex}, \mathbf{f}_2^{ex}, \dots, \mathbf{f}_{N_{ex}}^{ex}\}$

:external-moment-list

[method]

return $\{\mathbf{n}_1^{ex}, \mathbf{n}_2^{ex}, \dots, \mathbf{n}_{N_{ex}}^{ex}\}$

:mg-vec

[method]

return $m\mathbf{g}$

:cog \mathcal{E}_{key} (update? t)

[method]

return $\mathbf{p}_G(\mathbf{q})$

:kinematics-task-value \mathcal{E}_{key} (update? t)

[method]

$$\mathbf{e}^{kin}(\mathbf{q}) = \mathbf{e}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \quad (31)$$

$$= \begin{pmatrix} \mathbf{e}_1^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \mathbf{e}_2^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \vdots \\ \mathbf{e}_{N_{kin}}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \end{pmatrix} \quad (32)$$

$$\mathbf{e}_m^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) = K_{kin} \begin{pmatrix} \mathbf{p}_m^{kin-trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \mathbf{p}_m^{kin-att}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \mathbf{a} \left(\mathbf{R}_m^{kin-trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) \mathbf{R}_m^{kin-att}(\boldsymbol{\theta}, \boldsymbol{\phi})^T \right) \end{pmatrix} \in \mathbb{R}^6 \quad (m = 1, 2, \dots, N_{kin}) \quad (33)$$

$a(R)$ は姿勢行列 R の等価角軸ベクトルを表す .

return $e^{kin}(q) \in \mathbb{R}^{6N_{kin}}$

:eom-trans-task-value $\mathcal{E}key$ (update? t)

[method]

$$e^{eom-trans}(q) = e^{eom-trans}(\hat{w}) \quad (34)$$

$$= \sum_{m=1}^{N_{cnt}} f_m - mg + \sum_{m=1}^{N_{ex}} f_m^{ex} \quad (35)$$

return $e^{eom-trans}(q) \in \mathbb{R}^3$

:eom-rot-task-value $\mathcal{E}key$ (update? t)

[method]

$$e^{eom-rot}(q) = e^{eom-rot}(\theta, \hat{w}, \phi) \quad (36)$$

$$= \sum_{m=1}^{N_{cnt}} \{ (p_m^{cnt-trg}(\theta, \phi) - p_G(\theta, \phi)) \times f_m + n_m \} \\ + \sum_{m=1}^{N_{ex}} \{ (p_m^{ex}(\theta, \phi) - p_G(\theta, \phi)) \times f_m^{ex} + n_m^{ex} \} \quad (37)$$

return $e^{eom-rot}(q) \in \mathbb{R}^3$

:torque-task-value $\mathcal{E}key$ (update? t)

[method]

$$e^{trq}(q) = e^{trq}(\theta, \hat{w}, \tau, \phi) \quad (38)$$

$$= \tau + \sum_{m=1}^{N_{cnt}} \tau_m^{cnt}(\theta, \phi) - \tau^{grav}(\theta, \phi) + \sum_{m=1}^{N_{ex}} \tau_m^{ex}(\theta, \phi) \quad (39)$$

$$= \tau + \sum_{m=1}^{N_{cnt}} J_{drive-joint,m}^{cnt-att}(\theta, \phi)^T w_m - \tau^{grav}(\theta, \phi) + \sum_{m=1}^{N_{ex}} J_{drive-joint,m}^{ex}(\theta, \phi)^T w_m^{ex} \quad (40)$$

$\tau_m^{cnt}(\theta, \phi)$ は m 番目の接触部位にかかる接触レンチ w_m による関節トルク , $\tau_m^{grav}(\theta, \phi)$ は自重による関節トルクを表す .

return $e^{trq}(q) \in \mathbb{R}^{N_{drive-joint}}$

:posture-task-value $\mathcal{E}key$ (update? t)

[method]

$$e^{posture}(q) = e^{posture}(\theta) \quad (41)$$

$$= k_{posture}(\bar{\theta}^{trg} - \bar{\theta}) \quad (42)$$

$\bar{\theta}^{trg}, \bar{\theta}$ は着目関節リスト $\mathcal{J}_{posture}$ の目標関節角と現在の関節角 .

return $e^{posture}(q) \in \mathbb{R}^{N_{posture-joint}}$

:task-value $\mathcal{E}key$ (update? t)

[method]

$$\text{return } e(q) := \begin{pmatrix} e^{kin}(q) \\ e^{eom-trans}(q) \\ e^{eom-rot}(q) \\ e^{trq}(q) \\ e^{posture}(q) \end{pmatrix} = \begin{pmatrix} e^{kin}(\theta, \phi) \\ e^{eom-trans}(\hat{w}) \\ e^{eom-rot}(\theta, \hat{w}, \phi) \\ e^{trq}(\theta, \hat{w}, \tau, \phi) \\ e^{posture}(\theta) \end{pmatrix}$$

:kinematics-task-jacobian-with-theta

[method]

$$\frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial \mathbf{e}_1^{kin}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mathbf{e}_2^{kin}}{\partial \boldsymbol{\theta}} \\ \vdots \\ \frac{\partial \mathbf{e}_{N_{kin}}^{kin}}{\partial \boldsymbol{\theta}} \end{pmatrix} \quad (43)$$

$$\frac{\partial \mathbf{e}_m^{kin}}{\partial \boldsymbol{\theta}} = K_{kin} \left\{ \mathbf{J}_{\theta,m}^{kin-trg}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{\theta,m}^{kin-att}(\boldsymbol{\theta}, \phi) \right\} \quad (m = 1, 2, \dots, N_{kin}) \quad (44)$$

$$\text{return } \frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{6N_{kin} \times N_{var-joint}}$$

:kinematics-task-jacobian-with-phi

[method]

$$\frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\phi}} = \begin{pmatrix} \frac{\partial \mathbf{e}_1^{kin}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \mathbf{e}_2^{kin}}{\partial \boldsymbol{\phi}} \\ \vdots \\ \frac{\partial \mathbf{e}_{N_{kin}}^{kin}}{\partial \boldsymbol{\phi}} \end{pmatrix} \quad (45)$$

$$\frac{\partial \mathbf{e}_m^{kin}}{\partial \boldsymbol{\phi}} = K_{kin} \left\{ \mathbf{J}_{\phi,m}^{kin-trg}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{\phi,m}^{kin-att}(\boldsymbol{\theta}, \phi) \right\} \quad (m = 1, 2, \dots, N_{kin}) \quad (46)$$

$$\text{return } \frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\phi}} \in \mathbb{R}^{6N_{kin} \times N_{invar-joint}}$$

:eom-trans-task-jacobian-with-wrench

[method]

$$\frac{\partial \mathbf{e}^{eom-trans}}{\partial \hat{\mathbf{w}}} = \begin{pmatrix} \frac{\partial \mathbf{e}^{eom-trans}}{\partial \mathbf{f}_1} & \frac{\partial \mathbf{e}^{eom-trans}}{\partial \mathbf{n}_1} & \dots & \frac{\partial \mathbf{e}^{eom-trans}}{\partial \mathbf{f}_{N_{cnt}}} & \frac{\partial \mathbf{e}^{eom-trans}}{\partial \mathbf{n}_{N_{cnt}}} \end{pmatrix} \quad (47)$$

$$= \begin{pmatrix} \mathbf{I}_3 & \mathbf{O}_3 & \dots & \mathbf{I}_3 & \mathbf{O}_3 \end{pmatrix} \quad (48)$$

$$\text{return } \frac{\partial \mathbf{e}^{eom-trans}}{\partial \hat{\mathbf{w}}} \in \mathbb{R}^{3 \times 6N_{cnt}}$$

:eom-rot-task-jacobian-with-theta

[method]

$$\begin{aligned} \frac{\partial \mathbf{e}^{eom-rot}}{\partial \boldsymbol{\theta}} &= \sum_{m=1}^{N_{cnt}} \left\{ -[\mathbf{f}_m \times] \left(\mathbf{J}_{\theta,m}^{cnt-trg}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{G\theta}(\boldsymbol{\theta}, \phi) \right) \right\} \\ &\quad + \sum_{m=1}^{N_{ex}} \left\{ -[\mathbf{f}_m^{ex} \times] \left(\mathbf{J}_{\theta,m}^{ex}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{G\theta}(\boldsymbol{\theta}, \phi) \right) \right\} \end{aligned} \quad (49)$$

$$\begin{aligned} &= \left[\left(\sum_{m=1}^{N_{cnt}} \mathbf{f}_m + \sum_{m=1}^{N_{ex}} \mathbf{f}_m^{ex} \right) \times \right] \mathbf{J}_{G\theta}(\boldsymbol{\theta}, \phi) \\ &\quad - \sum_{m=1}^{N_{cnt}} [\mathbf{f}_m \times] \mathbf{J}_{\theta,m}^{cnt-trg}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{ex}} [\mathbf{f}_m^{ex} \times] \mathbf{J}_{\theta,m}^{ex}(\boldsymbol{\theta}, \phi) \end{aligned} \quad (50)$$

$\sum_{m=1}^{N_{cnt}} \mathbf{f}_m + \sum_{m=1}^{N_{ex}} \mathbf{f}_m^{ex} = m\mathbf{g}$ つまり, eom-trans-task が成立すると仮定すると次式が成り立つ.

$$\frac{\partial \mathbf{e}^{eom-rot}}{\partial \boldsymbol{\theta}} = [m\mathbf{g} \times] \mathbf{J}_{G\theta}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{cnt}} [\mathbf{f}_m \times] \mathbf{J}_{\theta,m}^{cnt-trg}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{ex}} [\mathbf{f}_m^{ex} \times] \mathbf{J}_{\theta,m}^{ex}(\boldsymbol{\theta}, \phi) \quad (51)$$

return $\frac{\partial \mathbf{e}^{eom-rot}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{3 \times N_{var-joint}}$

:eom-rot-task-jacobian-with-wrench

[method]

$$\frac{\partial \mathbf{e}^{eom-rot}}{\partial \hat{\mathbf{w}}} = \begin{pmatrix} \frac{\partial \mathbf{e}^{eom-rot}}{\partial \mathbf{f}_1} & \frac{\partial \mathbf{e}^{eom-rot}}{\partial \mathbf{n}_1} & \cdots & \frac{\partial \mathbf{e}^{eom-rot}}{\partial \mathbf{f}_{N_{cnt}}} & \frac{\partial \mathbf{e}^{eom-rot}}{\partial \mathbf{n}_{N_{cnt}}} \end{pmatrix} \quad (52)$$

$$\frac{\partial \mathbf{e}^{eom-rot}}{\partial \mathbf{f}_m} = [(\mathbf{p}_m^{cnt-trg}(\boldsymbol{\theta}, \phi) - \mathbf{p}_G(\boldsymbol{\theta}, \phi)) \times] \quad (m = 1, 2, \dots, N_{cnt}) \quad (53)$$

$$\frac{\partial \mathbf{e}^{eom-rot}}{\partial \mathbf{n}_m} = \mathbf{I}_3 \quad (m = 1, 2, \dots, N_{cnt}) \quad (54)$$

return $\frac{\partial \mathbf{e}^{eom-rot}}{\partial \hat{\mathbf{w}}} \in \mathbb{R}^{3 \times 6N_{cnt}}$

:eom-rot-task-jacobian-with-phi

[method]

$$\begin{aligned} \frac{\partial \mathbf{e}^{eom-rot}}{\partial \phi} &= \sum_{m=1}^{N_{cnt}} \left\{ -[\mathbf{f}_m \times] \left(\mathbf{J}_{\phi, m}^{cnt-trg}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{G\phi}(\boldsymbol{\theta}, \phi) \right) \right\} \\ &\quad + \sum_{m=1}^{N_{ex}} \left\{ -[\mathbf{f}_m^{ex} \times] \left(\mathbf{J}_{\phi, m}^{ex}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{G\phi}(\boldsymbol{\theta}, \phi) \right) \right\} \end{aligned} \quad (55)$$

$$\begin{aligned} &= \left[\left(\sum_{m=1}^{N_{cnt}} \mathbf{f}_m + \sum_{m=1}^{N_{ex}} \mathbf{f}_m^{ex} \right) \times \right] \mathbf{J}_{G\phi}(\boldsymbol{\theta}, \phi) \\ &\quad - \sum_{m=1}^{N_{cnt}} [\mathbf{f}_m \times] \mathbf{J}_{\phi, m}^{cnt-trg}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{ex}} [\mathbf{f}_m^{ex} \times] \mathbf{J}_{\phi, m}^{ex}(\boldsymbol{\theta}, \phi) \end{aligned} \quad (56)$$

$\sum_{m=1}^{N_{cnt}} \mathbf{f}_m + \sum_{m=1}^{N_{ex}} \mathbf{f}_m^{ex} = m\mathbf{g}$ つまり, eom-trans-task が成立すると仮定すると次式が成り立つ.

$$\frac{\partial \mathbf{e}^{eom-rot}}{\partial \phi} = [m\mathbf{g} \times] \mathbf{J}_{G\phi}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{cnt}} [\mathbf{f}_m \times] \mathbf{J}_{\phi, m}^{cnt-trg}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{ex}} [\mathbf{f}_m^{ex} \times] \mathbf{J}_{\phi, m}^{ex}(\boldsymbol{\theta}, \phi) \quad (57)$$

return $\frac{\partial \mathbf{e}^{eom-rot}}{\partial \phi} \in \mathbb{R}^{3 \times N_{invar-joint}}$

:torque-task-jacobian-with-theta

[method]

$$\frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\theta}} = \sum_{m=1}^{N_{cnt}} \frac{\partial \boldsymbol{\tau}_m^{cnt}}{\partial \boldsymbol{\theta}} - \frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}} + \sum_{m=1}^{N_{ex}} \frac{\partial \boldsymbol{\tau}_m^{ex}}{\partial \boldsymbol{\theta}} \quad (58)$$

return $\frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{N_{drive-joint} \times N_{var-joint}}$

:torque-task-jacobian-with-wrench

[method]

$$\frac{\partial \mathbf{e}^{trq}}{\partial \hat{\mathbf{w}}} = \begin{pmatrix} \frac{\partial \mathbf{e}^{trq}}{\partial \mathbf{w}_1} & \frac{\partial \mathbf{e}^{trq}}{\partial \mathbf{w}_2} & \cdots & \frac{\partial \mathbf{e}^{trq}}{\partial \mathbf{w}_{N_{cnt}}} \end{pmatrix} \quad (59)$$

$$\frac{\partial \mathbf{e}^{trq}}{\partial \mathbf{w}_m} = \mathbf{J}_{drive-joint, m}^{cnt-att}(\boldsymbol{\theta}, \phi)^T \quad (m = 1, 2, \dots, N_{cnt}) \quad (60)$$

return $\frac{\partial \mathbf{e}^{trq}}{\partial \hat{\mathbf{w}}} \in \mathbb{R}^{N_{drive-joint} \times 6N_{cnt}}$

:torque-task-jacobian-with-phi

[method]

$$\frac{\partial \mathbf{e}^{trq}}{\partial \phi} = \sum_{m=1}^{N_{cnt}} \frac{\partial \tau_m^{cnt}}{\partial \phi} - \frac{\partial \tau_m^{grav}}{\partial \phi} + \sum_{m=1}^{N_{ex}} \frac{\partial \tau_m^{ex}}{\partial \phi} \quad (61)$$

$$\text{return } \frac{\partial \mathbf{e}^{trq}}{\partial \phi} \in \mathbb{R}^{N_{drive-joint} \times N_{invar-joint}}$$

:torque-task-jacobian-with-torque

[method]

$$\frac{\partial \mathbf{e}^{trq}}{\partial \tau} = \mathbf{I}_{N_{drive-joint}} \quad (62)$$

$$\text{return } \frac{\partial \mathbf{e}^{trq}}{\partial \tau} \in \mathbb{R}^{N_{drive-joint} \times N_{drive-joint}}$$

:posture-task-jacobian-with-theta *key (update? nil)*

[method]

$$\left(\frac{\partial \mathbf{e}^{posture}}{\partial \theta} \right)_{i,j} = \begin{cases} -k_{posture} & (\mathcal{J}_{posture,i} = \mathcal{J}_{var,j}) \\ 0 & \text{otherwise} \end{cases} \quad (63)$$

$$\text{return } \frac{\partial \mathbf{e}^{posture}}{\partial \theta} \in \mathbb{R}^{N_{posture-joint} \times N_{var-joint}}$$

:variant-task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} = \begin{matrix} 6N_{kin} \\ 3 \\ 3 \\ N_{drive-joint} \\ N_{posture-joint} \end{matrix} \begin{pmatrix} N_{var-joint} & 6N_{cnt} & N_{drive-joint} \\ \frac{\partial \mathbf{e}^{kin}}{\partial \theta} & \frac{\partial \mathbf{e}^{com-trans}}{\partial \hat{\mathbf{w}}} & \\ \frac{\partial \mathbf{e}^{com-rot}}{\partial \theta} & \frac{\partial \mathbf{e}^{com-rot}}{\partial \hat{\mathbf{w}}} & \\ \frac{\partial \mathbf{e}^{trq}}{\partial \theta} & \frac{\partial \mathbf{e}^{trq}}{\partial \hat{\mathbf{w}}} & \frac{\partial \mathbf{e}^{trq}}{\partial \tau} \\ \frac{\partial \mathbf{e}^{posture}}{\partial \theta} & & \end{pmatrix} \quad (64)$$

$$\text{return } \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} \in \mathbb{R}^{(6N_{kin}+3+3+N_{drive-joint}+N_{posture-joint}) \times (N_{var-joint}+6N_{cnt}+N_{drive-joint})}$$

:invariant-task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} = \begin{matrix} 6N_{kin} \\ 3 \\ 3 \\ N_{drive-joint} \\ N_{posture-joint} \end{matrix} \begin{pmatrix} N_{invar-joint} \\ \frac{\partial \mathbf{e}^{kin}}{\partial \phi} \\ \frac{\partial \mathbf{e}^{com-rot}}{\partial \phi} \\ \frac{\partial \mathbf{e}^{trq}}{\partial \phi} \\ \end{pmatrix} \quad (65)$$

$$\text{return } \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} \in \mathbb{R}^{(6N_{kin}+3+3+N_{drive-joint}+N_{posture-joint}) \times N_{invar-joint}}$$

:task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}} = \begin{pmatrix} \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} & \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} \end{pmatrix} \quad (66)$$

$$= \begin{matrix} & N_{var-joint} & 6N_{cnt} & N_{drive-joint} & N_{invar-joint} \\ 6N_{kin} & \left(\frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\theta}} & & & \frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\phi}} \right) \\ 3 & & \frac{\partial \mathbf{e}^{com-trans}}{\partial \hat{\mathbf{w}}} & & \\ 3 & \frac{\partial \mathbf{e}^{com-rot}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{e}^{com-rot}}{\partial \hat{\mathbf{w}}} & & \frac{\partial \mathbf{e}^{com-rot}}{\partial \boldsymbol{\phi}} \\ N_{drive-joint} & \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{e}^{trq}}{\partial \hat{\mathbf{w}}} & \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\tau}} & \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\phi}} \\ N_{posture-joint} & \frac{\partial \mathbf{e}^{posture}}{\partial \boldsymbol{\theta}} & & & \end{matrix} \quad (67)$$

return $\frac{\partial \mathbf{e}}{\partial \mathbf{q}} \in \mathbb{R}^{(6N_{kin}+3+3+N_{drive-joint}+N_{posture-joint}) \times (N_{var-joint}+6N_{cnt}+N_{drive-joint}+N_{invar-joint})}$

:theta-max-vector $\mathcal{E}key$ (update? nil) [method]

return $\boldsymbol{\theta}_{max} \in \mathbb{R}^{N_{var-joint}}$

:theta-min-vector $\mathcal{E}key$ (update? nil) [method]

return $\boldsymbol{\theta}_{min} \in \mathbb{R}^{N_{var-joint}}$

:delta-theta-limit-vector $\mathcal{E}key$ (update? nil) [method]

get trust region of $\boldsymbol{\theta}$

return $\Delta \boldsymbol{\theta}_{limit}$

:theta-inequality-constraint-matrix $\mathcal{E}key$ (update? nil) [method]

$$\begin{cases} \boldsymbol{\theta}_{min} \leq \boldsymbol{\theta} + \Delta \boldsymbol{\theta} \leq \boldsymbol{\theta}_{max} \\ -\Delta \boldsymbol{\theta}_{limit} \leq \Delta \boldsymbol{\theta} \leq \Delta \boldsymbol{\theta}_{limit} \end{cases} \quad (\text{if } \Delta \boldsymbol{\theta}_{limit} \text{ is set}) \quad (68)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \Delta \boldsymbol{\theta} \geq \begin{pmatrix} \boldsymbol{\theta}_{min} - \boldsymbol{\theta} \\ -(\boldsymbol{\theta}_{max} - \boldsymbol{\theta}) \\ -\Delta \boldsymbol{\theta}_{limit} \\ -\Delta \boldsymbol{\theta}_{limit} \end{pmatrix} \quad (69)$$

$$\Leftrightarrow \mathbf{C}_{\boldsymbol{\theta}} \Delta \boldsymbol{\theta} \geq \mathbf{d}_{\boldsymbol{\theta}} \quad (70)$$

$$\text{return } \mathbf{C}_{\boldsymbol{\theta}} := \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \in \mathbb{R}^{4N_{var-joint} \times N_{var-joint}}$$

:theta-inequality-constraint-vector $\mathcal{E}key$ (update? t) [method]

$$\text{return } \mathbf{d}_{\boldsymbol{\theta}} := \begin{pmatrix} \boldsymbol{\theta}_{min} - \boldsymbol{\theta} \\ -(\boldsymbol{\theta}_{max} - \boldsymbol{\theta}) \\ -\Delta \boldsymbol{\theta}_{limit} \\ -\Delta \boldsymbol{\theta}_{limit} \end{pmatrix} \in \mathbb{R}^{4N_{var-joint}}$$

:wrench-inequality-constraint-matrix $\mathcal{E}key$ (update? t) [method]

接触レンチ $\mathbf{w} \in \mathbb{R}^6$ が満たすべき制約（非負制約，摩擦制約，圧力中心制約）が次式のように表されるとする．

$$\mathbf{C}_w \mathbf{w} \geq \mathbf{d}_w \quad (71)$$

N_{cnt} 箇所の接触部位の接触レンチを並べたベクトル \hat{w} の不等式制約は次式で表される .

$$C_{w,m}(\mathbf{w}_m + \Delta \mathbf{w}_m) \geq \mathbf{d}_{w,m} \quad (m = 1, 2, \dots, N_{cnt}) \quad (72)$$

$$\Leftrightarrow C_{w,m} \Delta \mathbf{w}_m \geq \mathbf{d}_{w,m} - C_{w,m} \mathbf{w}_m \quad (m = 1, 2, \dots, N_{cnt}) \quad (73)$$

$$\Leftrightarrow \begin{pmatrix} C_{w,1} & & & \\ & C_{w,2} & & \\ & & \ddots & \\ & & & C_{w,N_{cnt}} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{w}_1 \\ \Delta \mathbf{w}_2 \\ \vdots \\ \Delta \mathbf{w}_{N_{cnt}} \end{pmatrix} \geq \begin{pmatrix} \mathbf{d}_{w,1} - C_{w,1} \mathbf{w}_1 \\ \mathbf{d}_{w,2} - C_{w,2} \mathbf{w}_2 \\ \vdots \\ \mathbf{d}_{w,N_{cnt}} - C_{w,N_{cnt}} \mathbf{w}_{N_{cnt}} \end{pmatrix} \quad (74)$$

$$\Leftrightarrow C_{\hat{w}} \Delta \hat{\mathbf{w}} \geq \mathbf{d}_{\hat{w}} \quad (75)$$

$$\text{return } C_{\hat{w}} := \begin{pmatrix} C_{w,1} & & & \\ & C_{w,2} & & \\ & & \ddots & \\ & & & C_{w,N_{cnt}} \end{pmatrix} \in \mathbb{R}^{N_{wrench-ineq} \times \dim(\hat{\mathbf{w}})}$$

:wrench-inequality-constraint-vector $\mathcal{E}key \text{ (update? } t)$ [method]

$$\text{return } \mathbf{d}_{\hat{w}} := \begin{pmatrix} \mathbf{d}_{w,1} - C_{w,1} \mathbf{w}_1 \\ \mathbf{d}_{w,2} - C_{w,2} \mathbf{w}_2 \\ \vdots \\ \mathbf{d}_{w,N_{cnt}} - C_{w,N_{cnt}} \mathbf{w}_{N_{cnt}} \end{pmatrix} \in \mathbb{R}^{N_{wrench-ineq}}$$

:torque-max-vector $\mathcal{E}key \text{ (update? nil)}$ [method]

$$\text{return } \boldsymbol{\tau}_{max} \in \mathbb{R}^{N_{drive-joint}}$$

:torque-min-vector $\mathcal{E}key \text{ (update? nil)}$ [method]

$$\text{return } \boldsymbol{\tau}_{min} \in \mathbb{R}^{N_{drive-joint}}$$

:torque-inequality-constraint-matrix $\mathcal{E}key \text{ (update? nil)}$ [method]

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau} + \Delta \boldsymbol{\tau} \leq \boldsymbol{\tau}_{max} \quad (76)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \Delta \boldsymbol{\tau} \geq \begin{pmatrix} \boldsymbol{\tau}_{min} - \boldsymbol{\tau} \\ -(\boldsymbol{\tau}_{max} - \boldsymbol{\tau}) \end{pmatrix} \quad (77)$$

$$\Leftrightarrow C_{\boldsymbol{\tau}} \Delta \boldsymbol{\tau} \geq \mathbf{d}_{\boldsymbol{\tau}} \quad (78)$$

$$\text{return } C_{\boldsymbol{\tau}} := \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \in \mathbb{R}^{2N_{drive-joint} \times N_{drive-joint}}$$

:torque-inequality-constraint-vector $\mathcal{E}key \text{ (update? } t)$ [method]

$$\text{return } \mathbf{d}_{\boldsymbol{\tau}} := \begin{pmatrix} \boldsymbol{\tau}_{min} - \boldsymbol{\tau} \\ -(\boldsymbol{\tau}_{max} - \boldsymbol{\tau}) \end{pmatrix} \in \mathbb{R}^{2N_{drive-joint}}$$

:phi-max-vector $\mathcal{E}key \text{ (update? nil)}$ [method]

$$\text{return } \boldsymbol{\phi}_{max} \in \mathbb{R}^{N_{invar-joint}}$$

:phi-min-vector $\mathcal{E}key \text{ (update? nil)}$ [method]

$$\text{return } \boldsymbol{\phi}_{min} \in \mathbb{R}^{N_{invar-joint}}$$

:delta-phi-limit-vector $\mathcal{E}key \text{ (update? nil)}$ [method]

get trust region of $\boldsymbol{\phi}$

$$\text{return } \Delta \boldsymbol{\phi}_{limit}$$

:phi-inequality-constraint-matrix $\mathcal{E}key$ (*update? nil*)

[method]

$$\begin{cases} \phi_{min} \leq \phi + \Delta\phi \leq \phi_{max} \\ -\Delta\phi_{limit} \leq \Delta\phi \leq \Delta\phi_{limit} \quad (\text{if } \Delta\phi_{limit} \text{ is set}) \end{cases} \quad (79)$$

$$\Leftrightarrow \begin{pmatrix} I \\ -I \\ I \\ -I \end{pmatrix} \Delta\phi \geq \begin{pmatrix} \phi_{min} - \phi \\ -(\phi_{max} - \phi) \\ -\Delta\phi_{limit} \\ -\Delta\phi_{limit} \end{pmatrix} \quad (80)$$

$$\Leftrightarrow C_\phi \Delta\phi \geq d_\phi \quad (81)$$

$$\text{return } C_\phi := \begin{pmatrix} I \\ -I \\ I \\ -I \end{pmatrix} \in \mathbb{R}^{4N_{invar-joint} \times N_{invar-joint}}$$

:phi-inequality-constraint-vector $\mathcal{E}key$ (*update? t*)

[method]

$$\text{return } d_\phi := \begin{pmatrix} \phi_{min} - \phi \\ -(\phi_{max} - \phi) \\ -\Delta\phi_{limit} \\ -\Delta\phi_{limit} \end{pmatrix} \in \mathbb{R}^{4N_{invar-joint}}$$

:variant-config-inequality-constraint-matrix $\mathcal{E}key$ (*update? nil*)

[method]

$$\begin{cases} C_\theta \Delta\theta \geq d_\theta \\ C_{\hat{w}} \Delta\hat{w} \geq d_{\hat{w}} \\ C_\tau \Delta\tau \geq d_\tau \end{cases} \quad (82)$$

$$\Leftrightarrow \begin{pmatrix} C_\theta & & \\ & C_{\hat{w}} & \\ & & C_\tau \end{pmatrix} \begin{pmatrix} \Delta\theta \\ \Delta\hat{w} \\ \Delta\tau \end{pmatrix} \geq \begin{pmatrix} d_\theta \\ d_{\hat{w}} \\ d_\tau \end{pmatrix} \quad (83)$$

$$\Leftrightarrow C_{var} \Delta q_{var} \geq d_{var} \quad (84)$$

$$\text{return } C_{var} := \begin{pmatrix} C_\theta & & \\ & C_{\hat{w}} & \\ & & C_\tau \end{pmatrix} \in \mathbb{R}^{N_{var-ineq} \times \dim(\mathbf{q}_{var})}$$

:variant-config-inequality-constraint-vector $\mathcal{E}key$ (*update? t*)

[method]

$$\text{return } d_{var} := \begin{pmatrix} d_\theta \\ d_{\hat{w}} \\ d_\tau \end{pmatrix} \in \mathbb{R}^{N_{var-ineq}}$$

:invariant-config-inequality-constraint-matrix $\mathcal{E}key$ (*update? nil*)

[method]

$$C_\phi \Delta\phi \geq d_\phi \quad (85)$$

$$\Leftrightarrow C_{invar} \Delta q_{invar} \geq d_{invar} \quad (86)$$

$$\text{return } C_{invar} := C_\phi \in \mathbb{R}^{N_{invar-ineq} \times \dim(\mathbf{q}_{invar})}$$

:invariant-config-inequality-constraint-vector $\mathcal{E}key$ (*update? t*)

[method]

$$\begin{cases} C_{var}\Delta\mathbf{q}_{var} \geq \mathbf{d}_{var} \\ C_{invar}\Delta\mathbf{q}_{invar} \geq \mathbf{d}_{invar} \\ C_{col}\begin{pmatrix} \Delta\mathbf{q}_{var} \\ \Delta\mathbf{q}_{invar} \end{pmatrix} \geq \mathbf{d}_{col} \end{cases} \quad (87)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{C}_{var} \\ \hline \mathbf{C}_{invar} \\ \hline \mathbf{C}_{col} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{q}_{var} \\ \Delta \mathbf{q}_{invar} \end{pmatrix} \geq \begin{pmatrix} \mathbf{d}_{var} \\ \mathbf{d}_{invar} \\ \mathbf{d}_{col} \end{pmatrix} \quad (88)$$

$$\Leftrightarrow C\Delta q \geq d \quad (89)$$

$$\text{return } C := \begin{pmatrix} C_{var} & \\ & C_{invar} \\ \hline & C_{col} \end{pmatrix} \in \mathbb{R}^{N_{ineq} \times \dim(\mathbf{q})}$$

$$\text{return } \mathbf{d} := \begin{pmatrix} \mathbf{d}_{var} \\ \mathbf{d}_{invar} \\ \mathbf{d}_{col} \end{pmatrix} \in \mathbb{R}^{N_{ineq}}$$

return $\mathbf{A}_{var} \in \mathbb{R}^{0 \times \dim(\mathbf{q}_{var})}$ (no equality constraint)

return $\mathbf{b}_{var} \in \mathbb{R}^0$ (no equality constraint)

return $\mathbf{A}_{invar} \in \mathbb{R}^{0 \times \dim(\mathbf{q}_{invar})}$ (no equality constraint)

return $\mathbf{b}_{invar} \in \mathbb{R}^0$ (no equality constraint)

return $\mathbf{A} \in \mathbb{R}^{0 \times \dim(\mathbf{q})}$ (no equality constraint)

return $\mathbf{b} \in \mathbb{R}^0$ (no equality constraint)

二次形式の正則化項として次式を考える.

$$F_{tau}(\mathbf{q}) = \left\| \frac{\boldsymbol{\tau}}{\tau_{max}} \right\|^2 \quad (\text{ベクトルの要素ごとの割り算を表す}) \quad (90)$$

$$= \tau^T \bar{W} \tau \quad (91)$$

ここで,

$$\bar{W}_\tau := \begin{pmatrix} \frac{1}{\tau_{max,1}^2} & & & \\ & \frac{1}{\tau_{max,2}^2} & & \\ & & \ddots & \\ & & & \frac{1}{\tau_{max,N_{drive-joint}}^2} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{\tau}) \times dim(\boldsymbol{\tau})} \quad (92)$$

only-variant? is true:

$$\mathbf{W}_\tau := \begin{matrix} & dim(\boldsymbol{\theta}) & dim(\hat{\mathbf{w}}) & dim(\boldsymbol{\tau}) \\ \begin{matrix} dim(\boldsymbol{\theta}) \\ dim(\hat{\mathbf{w}}) \\ dim(\boldsymbol{\tau}) \end{matrix} & \begin{pmatrix} & & \\ & & \\ & & \bar{W}_\tau \end{pmatrix} \end{matrix} \in \mathbb{R}^{dim(\mathbf{q}_{var}) \times dim(\mathbf{q}_{var})} \quad (93)$$

otherwise:

$$\mathbf{W}_\tau := \begin{matrix} & dim(\boldsymbol{\theta}) & dim(\hat{\mathbf{w}}) & dim(\boldsymbol{\tau}) & dim(\boldsymbol{\phi}) \\ \begin{matrix} dim(\boldsymbol{\theta}) \\ dim(\hat{\mathbf{w}}) \\ dim(\boldsymbol{\tau}) \\ dim(\boldsymbol{\phi}) \end{matrix} & \begin{pmatrix} & & & \\ & & & \\ & & \bar{W}_\tau & \\ & & & \end{pmatrix} \end{matrix} \in \mathbb{R}^{dim(\mathbf{q}) \times dim(\mathbf{q})} \quad (94)$$

return \mathbf{W}_τ

:torque-regular-vector $\mathcal{E}key$ (*update? t*) [method]
(*only-variant? nil*)

$$\bar{\mathbf{v}}_\tau := \bar{W}_\tau \boldsymbol{\tau} \quad (95)$$

$$= \begin{pmatrix} \frac{\tau_1}{\tau_{max,1}^2} \\ \frac{\tau_2}{\tau_{max,2}^2} \\ \vdots \\ \frac{\tau_{dim(\boldsymbol{\tau})}}{\tau_{max,dim(\boldsymbol{\tau})}^2} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{\tau})} \quad (96)$$

only-variant? is true:

$$\mathbf{v}_\tau := \begin{matrix} & 1 \\ \begin{matrix} dim(\boldsymbol{\theta}) \\ dim(\hat{\mathbf{w}}) \\ dim(\boldsymbol{\tau}) \end{matrix} & \begin{pmatrix} \\ \\ \bar{\mathbf{v}}_\tau \end{pmatrix} \end{matrix} \in \mathbb{R}^{dim(\mathbf{q}_{var})} \quad (97)$$

otherwise:

$$\mathbf{v}_\tau := \begin{matrix} & 1 \\ \begin{matrix} dim(\boldsymbol{\theta}) \\ dim(\hat{\mathbf{w}}) \\ dim(\boldsymbol{\tau}) \\ dim(\boldsymbol{\phi}) \end{matrix} & \begin{pmatrix} \\ \\ \bar{\mathbf{v}}_\tau \\ \end{pmatrix} \end{matrix} \in \mathbb{R}^{dim(\mathbf{q})} \quad (98)$$

return \mathbf{v}_τ

:torque-ratio

[method]

$$\text{return } \frac{\boldsymbol{\tau}}{\tau_{max}} := \begin{pmatrix} \frac{\tau_1}{\tau_{max,1}} \\ \frac{\tau_2}{\tau_{max,2}} \\ \vdots \\ \frac{\tau_{N_{drive-joint}}}{\tau_{max,N_{drive-joint}}} \end{pmatrix}$$

:regular-matrix

[method]

$$\mathbf{W}_{reg} := \min(k_{max}, \|\mathbf{e}\|^2 + k_{off})\mathbf{I} + \mathbf{W}_{\tau} \quad (99)$$

$$\text{return } \mathbf{W}_{reg} \in \mathbb{R}^{dim(\mathbf{q}) \times dim(\mathbf{q})}$$

:regular-vector

[method]

$$\mathbf{v}_{reg} := \mathbf{v}_{\tau} \quad (100)$$

$$\text{return } \mathbf{v}_{reg} \in \mathbb{R}^{dim(\mathbf{q})}$$

:update-collision-inequality-constraint

[method]

リンク 1 とリンク 2 の最近点を $\mathbf{p}_1, \mathbf{p}_2$ とする．リンク 1 とリンク 2 が干渉しない条件を，最近点 $\mathbf{p}_1, \mathbf{p}_2$ の距離が d_{margin} 以上である条件に置き換えて考える．これは次式で表される．

$$\mathbf{d}_{12}^T(\mathbf{p}_1 - \mathbf{p}_2) \geq d_{margin} \quad (101)$$

$$\text{where } \mathbf{d}_{12} = \mathbf{p}_1 - \mathbf{p}_2 \quad (102)$$

コンフィギュレーションが $\Delta \mathbf{q}$ だけ更新されてもこれが成立するための条件は次式で表される．

$$\mathbf{d}_{12}^T \{(\mathbf{p}_1 + \Delta \mathbf{p}_1) - (\mathbf{p}_2 + \Delta \mathbf{p}_2)\} \geq d_{margin} \quad (103)$$

$$\text{where } \Delta \mathbf{p}_1 = \mathbf{J}_{\theta,1} \Delta \boldsymbol{\theta} + \mathbf{J}_{\phi,1} \Delta \boldsymbol{\phi} \quad (104)$$

$$\Delta \mathbf{p}_2 = \mathbf{J}_{\theta,2} \Delta \boldsymbol{\theta} + \mathbf{J}_{\phi,2} \Delta \boldsymbol{\phi} \quad (105)$$

$$\mathbf{J}_{\theta,i} = \frac{\partial \mathbf{p}_i}{\partial \boldsymbol{\theta}}, \quad \mathbf{J}_{\phi,i} = \frac{\partial \mathbf{p}_i}{\partial \boldsymbol{\phi}} \quad (i = 1, 2) \quad (106)$$

これは以下のように変形される．

$$\mathbf{d}_{12}^T \{(\mathbf{p}_1 + \mathbf{J}_{\theta,1} \Delta \boldsymbol{\theta} + \mathbf{J}_{\phi,1} \Delta \boldsymbol{\phi}) - (\mathbf{p}_2 + \mathbf{J}_{\theta,2} \Delta \boldsymbol{\theta} + \mathbf{J}_{\phi,2} \Delta \boldsymbol{\phi})\} \geq d_{margin} \quad (107)$$

$$\Leftrightarrow \mathbf{d}_{12}^T (\mathbf{J}_{\theta,1} - \mathbf{J}_{\theta,2}) \Delta \boldsymbol{\theta} + \mathbf{d}_{12}^T (\mathbf{J}_{\phi,1} - \mathbf{J}_{\phi,2}) \Delta \boldsymbol{\phi} \geq -(\mathbf{d}_{12}^T (\mathbf{p}_1 - \mathbf{p}_2) - d_{margin}) \quad (108)$$

$$\Leftrightarrow \mathbf{c}_{col,var}^T \Delta \boldsymbol{\theta} + \mathbf{c}_{col,invar}^T \Delta \boldsymbol{\phi} \geq d_{col} \quad (109)$$

$$\text{where } \mathbf{c}_{col,var}^T = \mathbf{d}_{12}^T (\mathbf{J}_{\theta,1} - \mathbf{J}_{\theta,2}) \quad (110)$$

$$\mathbf{c}_{col,invar}^T = \mathbf{d}_{12}^T (\mathbf{J}_{\phi,1} - \mathbf{J}_{\phi,2}) \quad (111)$$

$$d_{col} = -(\mathbf{d}_{12}^T (\mathbf{p}_1 - \mathbf{p}_2) - d_{margin}) \quad (112)$$

i 番目の干渉回避リンクペアに関する行列，ベクトルをそれぞれ $\mathbf{c}_{col,var,i}^T, \mathbf{c}_{col,invar,i}^T, d_{col,i}$ とする． $i =$

$1, 2, \dots, N_{col}$ の全てのリンクペアにおいて干渉が生じないための条件は次式で表される .

$$\begin{pmatrix} \mathbf{C}_{col,\theta} & \mathbf{C}_{col,\phi} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\theta} \\ \Delta\boldsymbol{\phi} \end{pmatrix} \geq \mathbf{d}_{col} \quad (113)$$

$$\mathbf{C}_{col,\theta} := \begin{pmatrix} \mathbf{c}_{col,var,1}^T \\ \vdots \\ \mathbf{c}_{col,var,N_{col}}^T \end{pmatrix} \in \mathbb{R}^{N_{col} \times \dim(\boldsymbol{\theta})} \quad (114)$$

$$\mathbf{C}_{col,\phi} := \begin{pmatrix} \mathbf{c}_{col,invar,1}^T \\ \vdots \\ \mathbf{c}_{col,invar,N_{col}}^T \end{pmatrix} \in \mathbb{R}^{N_{col} \times \dim(\boldsymbol{\phi})}, \quad (115)$$

$$\mathbf{d}_{col} := \begin{pmatrix} d_{col,1} \\ \vdots \\ d_{col,N_{col}} \end{pmatrix} \in \mathbb{R}^{N_{col}} \quad (116)$$

update inequality matrix $\mathbf{C}_{col,\theta}, \mathbf{C}_{col,\phi}$ and inequality vector \mathbf{d}_{col} for collision avoidance

:collision-theta-inequality-constraint-matrix [method]

return $\mathbf{C}_{col,\theta} \in \mathbb{R}^{N_{col} \times \dim(\boldsymbol{\theta})}$

:collision-phi-inequality-constraint-matrix [method]

return $\mathbf{C}_{col,\phi} \in \mathbb{R}^{N_{col} \times \dim(\boldsymbol{\phi})}$

:collision-inequality-constraint-matrix *key (update? nil)* [method]

$$\mathbf{C}_{col} := N_{col} \begin{pmatrix} \dim(\boldsymbol{\theta}) & \dim(\hat{\mathbf{w}}) & \dim(\boldsymbol{\tau}) & \dim(\boldsymbol{\phi}) \\ \mathbf{C}_{col,\theta} & \mathbf{O} & \mathbf{O} & \mathbf{C}_{col,\phi} \end{pmatrix} \quad (117)$$

return $\mathbf{C}_{col} \in \mathbb{R}^{N_{col} \times \dim(\mathbf{q})}$

:collision-inequality-constraint-vector *key (update? nil)* [method]

return $\mathbf{d}_{col} \in \mathbb{R}^{N_{col}}$

:update-viewer [method]

Update viewer.

:print-status [method]

Print status.

2.2 軌道コンフィギュレーションと軌道タスク関数

trajectory-configuration-task [class]

:super **propertied-object**

:slots (*_instant-config-task-list* list of instant-config-task instance)

(*_num-instant-config-task* L)

(*_dim-variant-config* $\dim(\mathbf{q}_{var})$)

(*_dim-invariant-config* $\dim(\mathbf{q}_{invar})$)

```

(dim-config  $\dim(\mathbf{q})$ )
(dim-task  $\dim(\mathbf{e})$ )
(norm-regular-scale-max  $k_{max}$ )
(norm-regular-scale-offset  $k_{off}$ )
(adjacent-regular-scale  $k_{adj}$ )
(torque-regular-scale  $k_{trq}$ )
(task-jacobi buffer for  $\frac{\partial \mathbf{e}}{\partial \mathbf{q}}$ )

```

軌道コンフィギュレーション \mathbf{q} と軌道タスク関数 $\mathbf{e}(\mathbf{q})$ のクラス .

以降では , 説明文やメソッド名で , “軌道” や “trajectory” を省略する .

コンフィギュレーション \mathbf{q} の取得・更新 , タスク関数 $\mathbf{e}(\mathbf{q})$ の取得 , タスク関数のヤコビ行列 $\frac{\partial \mathbf{e}(\mathbf{q})}{\partial \mathbf{q}}$ の取得 , コンフィギュレーションの等式・不等式制約 A, b, C, d の取得のためのメソッドが定義されている .

コンフィギュレーション・タスク関数を定めるために , 初期化時に以下を与える

- 瞬時のコンフィギュレーション・タスクのリスト

`instant-config-task-list` `instant-configuration-task` のリスト

- 目的関数の重み

`norm-regular-scale-max` k_{max} コンフィギュレーション更新量正則化の重み最大値

`norm-regular-scale-offset` k_{off} コンフィギュレーション更新量正則化の重みオフセット

`adjacent-regular-scale` k_{adj} 隣接コンフィギュレーション正則化の重み

`torque-regular-scale` k_{trq} トルク正則化の重み

コンフィギュレーション \mathbf{q} は以下から構成される .

$$\mathbf{q} := \left(\mathbf{q}_{var}^{(1)T} \quad \mathbf{q}_{var}^{(2)T} \quad \cdots \quad \mathbf{q}_{var}^{(L)T} \quad \mathbf{q}_{invar}^T \right)^T \quad (118)$$

ここで ,

$$\mathbf{q}_{invar} := \mathbf{q}_{invar}^{(1)} = \mathbf{q}_{invar}^{(2)} = \cdots = \mathbf{q}_{invar}^{(L)} \quad (119)$$

$\mathbf{q}_{var}^{(l)}, \mathbf{q}_{invar}^{(l)}$ ($l = 1, 2, \dots, L$) は l 番目の瞬時の時変 , 時不変コンフィギュレーションを表す .

タスク関数 $\mathbf{e}(\mathbf{q})$ は以下から構成される .

$$\mathbf{e}(\mathbf{q}) := \left(\mathbf{e}^{(1)T}(\mathbf{q}_{var}^{(1)}, \mathbf{q}_{invar}^{(1)}) \quad \mathbf{e}^{(2)T}(\mathbf{q}_{var}^{(2)}, \mathbf{q}_{invar}^{(2)}) \quad \cdots \quad \mathbf{e}^{(L)T}(\mathbf{q}_{var}^{(L)}, \mathbf{q}_{invar}^{(L)}) \right)^T \quad (120)$$

$\mathbf{e}^{(l)}(\mathbf{q}_{var}^{(l)}, \mathbf{q}_{invar}^{(l)})$ ($l = 1, 2, \dots, L$) は l 番目の瞬時のタスク関数を表す .

:init *key* (*name*) [method]

(*instant-config-task-list*)

(*norm-regular-scale-max* 1.000000e-04)

(*norm-regular-scale-offset* 1.000000e-07)

(*adjacent-regular-scale* 0.005)

(*torque-regular-scale* 0.001)

Initialize instance

:instant-config-task-list [method]

return instant-config-task-list

:dim-variant-config [method]
 return $\dim(\mathbf{q}_{var}) := \sum_{l=1}^L \dim(\mathbf{q}_{var}^{(l)})$

:dim-invariant-config [method]
 return $\dim(\mathbf{q}_{invar}) := \dim(\mathbf{q}_{invar}^{(l)})$ ($l = 1, 2, \dots, L$ で同じ)

:dim-config [method]
 return $\dim(\mathbf{q}) := \dim(\mathbf{q}_{var}) + \dim(\mathbf{q}_{invar})$

:dim-task [method]
 return $\dim(\mathbf{e}) := \sum_{l=1}^L \dim(\mathbf{e}^{(l)})$

:variant-config-vector [method]
 return $\mathbf{q}_{var} := \begin{pmatrix} \mathbf{q}_{var}^{(1)} \\ \mathbf{q}_{var}^{(2)} \\ \vdots \\ \mathbf{q}_{var}^{(L)} \end{pmatrix}$

:invariant-config-vector [method]
 return $\mathbf{q}_{invar} := \mathbf{q}_{invar}^{(l)}$ ($l = 1, 2, \dots, L$ で同じ)

:config-vector [method]
 return $\mathbf{q} := \begin{pmatrix} \mathbf{q}_{var} \\ \mathbf{q}_{invar} \end{pmatrix} = \begin{pmatrix} \mathbf{q}_{var}^{(1)} \\ \mathbf{q}_{var}^{(2)} \\ \vdots \\ \mathbf{q}_{var}^{(L)} \\ \mathbf{q}_{invar} \end{pmatrix}$

:set-variant-config *variant-config-new* *ℰkey* (*relative?* *nil*) [method]
 (*apply-to-robot?* *t*)

Set \mathbf{q}_{var} .

:set-invariant-config *invariant-config-new* *ℰkey* (*relative?* *nil*) [method]
 (*apply-to-robot?* *t*)

Set \mathbf{q}_{invar} .

:set-config *config-new* *ℰkey* (*relative?* *nil*) [method]
 (*apply-to-robot?* *t*)

Set \mathbf{q} .

:task-value *ℰkey* (*update?* *t*) [method]
 return $\mathbf{e}(\mathbf{q}) := \begin{pmatrix} \mathbf{e}^{(1)}(\mathbf{q}_{var}, \mathbf{q}_{invar}) \\ \mathbf{e}^{(2)}(\mathbf{q}_{var}, \mathbf{q}_{invar}) \\ \vdots \\ \mathbf{e}^{(L)}(\mathbf{q}_{var}, \mathbf{q}_{invar}) \end{pmatrix}$

:variant-task-jacobian [method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} = \begin{pmatrix} \frac{\partial \mathbf{e}^{(1)}}{\partial \mathbf{q}_{var}^{(1)}} & & \mathbf{O} \\ & \frac{\partial \mathbf{e}^{(2)}}{\partial \mathbf{q}_{var}^{(2)}} & \\ & & \ddots \\ \mathbf{O} & & & \frac{\partial \mathbf{e}^{(L)}}{\partial \mathbf{q}_{var}^{(L)}} \end{pmatrix} \quad (121)$$

return $\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} \in \mathbb{R}^{dim(\mathbf{e}) \times dim(\mathbf{q}_{var})}$

:invariant-task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} = \begin{pmatrix} \frac{\partial \mathbf{e}^{(1)}}{\partial \mathbf{q}_{invar}^{(1)}} \\ \frac{\partial \mathbf{e}^{(2)}}{\partial \mathbf{q}_{invar}^{(2)}} \\ \vdots \\ \frac{\partial \mathbf{e}^{(L)}}{\partial \mathbf{q}_{invar}^{(L)}} \end{pmatrix} \quad (122)$$

return $\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} \in \mathbb{R}^{dim(\mathbf{e}) \times dim(\mathbf{q}_{invar})}$

:task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}} = \begin{pmatrix} \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} & \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} \end{pmatrix} \quad (123)$$

$$= \begin{pmatrix} \frac{\partial \mathbf{e}^{(1)}}{\partial \mathbf{q}_{var}^{(1)}} & & \mathbf{O} & \frac{\partial \mathbf{e}^{(1)}}{\partial \mathbf{q}_{invar}^{(1)}} \\ & \frac{\partial \mathbf{e}^{(2)}}{\partial \mathbf{q}_{var}^{(2)}} & & \frac{\partial \mathbf{e}^{(2)}}{\partial \mathbf{q}_{invar}^{(2)}} \\ & & \ddots & \\ \mathbf{O} & & & \frac{\partial \mathbf{e}^{(L)}}{\partial \mathbf{q}_{var}^{(L)}} & \frac{\partial \mathbf{e}^{(L)}}{\partial \mathbf{q}_{invar}^{(L)}} \end{pmatrix} \quad (124)$$

return $\frac{\partial \mathbf{e}}{\partial \mathbf{q}} \in \mathbb{R}^{dim(\mathbf{e}) \times dim(\mathbf{q})}$

:variant-config-inequality-constraint-matrix $\mathcal{E}key$ (*update? nil*)

[method]

$$\mathbf{C}_{var} := \begin{pmatrix} \mathbf{C}_{var}^{(1)} & & \mathbf{O} \\ & \mathbf{C}_{var}^{(2)} & \\ & & \ddots \\ \mathbf{O} & & & \mathbf{C}_{var}^{(L)} \end{pmatrix} \quad (125)$$

return $\mathbf{C}_{var} \in \mathbb{R}^{N_{var-ineq} \times dim(\mathbf{q}_{var})}$

:variant-config-inequality-constraint-vector $\mathcal{E}key$ (*update? t*)

[method]

$$\mathbf{d}_{var} := \begin{pmatrix} \mathbf{d}_{var}^{(1)} \\ \mathbf{d}_{var}^{(2)} \\ \vdots \\ \mathbf{d}_{var}^{(L)} \end{pmatrix} \quad (126)$$

return $\mathbf{d}_{var} \in \mathbb{R}^{N_{var-ineq}}$

:invariant-config-inequality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

$$\mathbf{C}_{invar} := \mathbf{C}_{invar}^{(l)} \quad (l = 1, 2, \dots, L \text{ で同じ}) \quad (127)$$

return $\mathbf{C}_{invar} \in \mathbb{R}^{N_{invar-ineq} \times \dim(\mathbf{q}_{invar})}$

:invariant-config-inequality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\mathbf{d}_{invar} := \mathbf{d}_{invar}^{(l)} \quad (l = 1, 2, \dots, L \text{ で同じ}) \quad (128)$$

return $\mathbf{d}_{invar} \in \mathbb{R}^{N_{invar-ineq}}$

:config-inequality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]
(*update-collision?* *nil*)

$$\mathbf{C} := \begin{pmatrix} \mathbf{C}_{var} & & \\ & \mathbf{C}_{invar} & \\ \cdots & & \\ & & \mathbf{C}_{col} \end{pmatrix} \in \mathbb{R}^{N_{ineq} \times \dim(\mathbf{q})} \quad (129)$$

return $\mathbf{C} \in \mathbb{R}^{N_{ineq} \times \dim(\mathbf{q})}$

:config-inequality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]
(*update-collision?* *nil*)

$$\mathbf{d} := \begin{pmatrix} \mathbf{d}_{var} \\ \mathbf{d}_{invar} \\ \mathbf{d}_{col} \end{pmatrix} \quad (130)$$

return $\mathbf{d} \in \mathbb{R}^{N_{ineq}}$

:variant-config-equality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

$$\mathbf{A}_{var} := \begin{pmatrix} \mathbf{A}_{var}^{(1)} & & & \mathbf{O} \\ & \mathbf{A}_{var}^{(2)} & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{A}_{var}^{(L)} \end{pmatrix} \quad (131)$$

return $\mathbf{A}_{var} \in \mathbb{R}^{N_{var-eq} \times \dim(\mathbf{q}_{var})}$

:variant-config-equality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\mathbf{b}_{var} := \begin{pmatrix} \mathbf{b}_{var}^{(1)} \\ \mathbf{b}_{var}^{(2)} \\ \vdots \\ \mathbf{b}_{var}^{(L)} \end{pmatrix} \quad (132)$$

return $\mathbf{b}_{var} \in \mathbb{R}^{N_{var-eq}}$

:invariant-config-equality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

$$\mathbf{A}_{invar} := \mathbf{A}_{invar}^{(l)} \quad (l = 1, 2, \dots, L \text{ で同じ}) \quad (133)$$

return $\mathbf{A}_{invar} \in \mathbb{R}^{N_{invar-eq} \times \dim(\mathbf{q}_{invar})}$

:invariant-config-equality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\mathbf{b}_{invar} := \mathbf{b}_{invar}^{(l)} \quad (l = 1, 2, \dots, L \text{ で同じ}) \quad (134)$$

return $\mathbf{b}_{invar} \in \mathbb{R}^{N_{invar-eq}}$

:config-equality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

$$\mathbf{A} := \begin{pmatrix} \mathbf{A}_{var} \\ \mathbf{A}_{invar} \end{pmatrix} \in \mathbb{R}^{N_{eq} \times \dim(\mathbf{q})} \quad (135)$$

return $\mathbf{A} \in \mathbb{R}^{N_{eq} \times \dim(\mathbf{q})}$

:config-equality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\mathbf{b} := \begin{pmatrix} \mathbf{b}_{var} \\ \mathbf{b}_{invar} \end{pmatrix} \quad (136)$$

return $\mathbf{b} \in \mathbb{R}^{N_{eq}}$

:update-collision-inequality-constraint [method]

update inequality matrix $\mathbf{C}_{col,\theta}^{(l)}$, $\mathbf{C}_{col,\phi}^{(l)}$ and inequality vector $\mathbf{d}_{col}^{(l)}$ for collision avoidance ($l = 1, 2, \dots, L$)

:collision-inequality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

$$\hat{\mathbf{C}}_{col,\theta}^{(l)} := N_{col}^{(l)} \begin{pmatrix} \dim(\boldsymbol{\theta}^{(l)}) & \dim(\hat{\mathbf{w}}^{(l)}) & \dim(\boldsymbol{\tau}^{(l)}) \\ \mathbf{C}_{col,\theta}^{(l)} & \mathbf{O} & \mathbf{O} \end{pmatrix} \quad (137)$$

$$\mathbf{C}_{col} := \begin{pmatrix} \hat{\mathbf{C}}_{col,\theta}^{(1)} & & & \mathbf{C}_{col,\phi}^{(1)} \\ & \hat{\mathbf{C}}_{col,\theta}^{(2)} & & \mathbf{C}_{col,\phi}^{(2)} \\ & & \ddots & \vdots \\ & & & \hat{\mathbf{C}}_{col,\theta}^{(L)} & \mathbf{C}_{col,\phi}^{(L)} \end{pmatrix} \quad (138)$$

return $\mathbf{C}_{col} \in \mathbb{R}^{N_{col} \times \dim(\mathbf{q})}$

:collision-inequality-constraint-vector $\mathcal{E}key$ (*update?* *nil*) [method]

$$\mathbf{d}_{col} := \begin{pmatrix} \mathbf{d}_{col}^{(1)} \\ \mathbf{d}_{col}^{(2)} \\ \vdots \\ \mathbf{d}_{col}^{(L)} \end{pmatrix} \quad (139)$$

return $\mathbf{d}_{col} \in \mathbb{R}^{N_{col}}$

:adjacent-regular-matrix $\mathcal{E}key$ (*update?* *nil*) [method]
 二次形式の正則化項として次式を考える .

$$F_{adj}(\mathbf{q}) = \sum_{l=1}^{L-1} \|\boldsymbol{\theta}_{l+1} - \boldsymbol{\theta}_l\|^2 \quad (140)$$

$$= \mathbf{q}^T \mathbf{W}_{adj} \mathbf{q} \quad (141)$$

ここで ,

$$\bar{\mathbf{I}}_{adj} := \begin{matrix} & \dim(\boldsymbol{\theta}^{(l)}) & \dim(\hat{\mathbf{w}}^{(l)}) & \dim(\boldsymbol{\tau}^{(l)}) \\ \begin{matrix} \dim(\boldsymbol{\theta}^{(l)}) \\ \dim(\hat{\mathbf{w}}^{(l)}) \\ \dim(\boldsymbol{\tau}^{(l)}) \end{matrix} & \begin{pmatrix} \mathbf{I} & & \\ & & \\ & & \end{pmatrix} \end{matrix} \in \mathbb{R}^{\dim(\mathbf{q}_{var}^{(l)}) \times \dim(\mathbf{q}_{var}^{(l)})} \quad (142)$$

$$\bar{\mathbf{W}}_{adj} := \begin{pmatrix} \bar{\mathbf{I}}_{adj} & -\bar{\mathbf{I}}_{adj} & & & & \\ -\bar{\mathbf{I}}_{adj} & 2\bar{\mathbf{I}}_{adj} & -\bar{\mathbf{I}}_{adj} & & & \mathbf{O} \\ & & \ddots & & & \\ & & & -\bar{\mathbf{I}}_{adj} & 2\bar{\mathbf{I}}_{adj} & -\bar{\mathbf{I}}_{adj} \\ \mathbf{O} & & & -\bar{\mathbf{I}}_{adj} & \bar{\mathbf{I}}_{adj} & \end{pmatrix} \in \mathbb{R}^{\dim(\mathbf{q}_{var}) \times \dim(\mathbf{q}_{var})} \quad (143)$$

$$\mathbf{W}_{adj} := \begin{pmatrix} \bar{\mathbf{W}}_{adj} & \\ & \mathbf{O} \end{pmatrix} \quad (144)$$

return $\mathbf{W}_{adj} \in \mathbb{R}^{\dim(\mathbf{q}) \times \dim(\mathbf{q})}$

:adjacent-regular-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\mathbf{v}_{adj} := \mathbf{W}_{adj} \mathbf{q} \quad (145)$$

return $\mathbf{v}_{adj} \in \mathbb{R}^{\dim(\mathbf{q})}$

:torque-regular-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

$$\bar{\mathbf{W}}_{\tau} := \begin{pmatrix} \mathbf{W}_{\tau}^{(1)} & & & \mathbf{O} \\ & \mathbf{W}_{\tau}^{(2)} & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{W}_{\tau}^{(L)} \end{pmatrix} \in \mathbb{R}^{\dim(\mathbf{q}_{var}) \times \dim(\mathbf{q}_{var})} \quad (146)$$

$$\mathbf{W}_{\tau} := \begin{pmatrix} \bar{\mathbf{W}}_{\tau} & \\ & \mathbf{O} \end{pmatrix} \quad (147)$$

return $\mathbf{W}_{\tau} \in \mathbb{R}^{\dim(\mathbf{q}) \times \dim(\mathbf{q})}$

:torque-regular-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\bar{\mathbf{v}}_{\tau} := \begin{pmatrix} \mathbf{v}_{\tau}^{(1)} \\ \mathbf{v}_{\tau}^{(2)} \\ \vdots \\ \mathbf{v}_{\tau}^{(L)} \end{pmatrix} \in \mathbb{R}^{\dim(\mathbf{q}_{var})} \quad (148)$$

$$\mathbf{v}_{\tau} := \begin{pmatrix} \bar{\mathbf{v}}_{\tau} \\ \mathbf{0} \end{pmatrix} \quad (149)$$

return $\mathbf{v}_\tau \in \mathbb{R}^{dim(\mathbf{q})}$

:regular-matrix [method]

$$\mathbf{W}_{reg} := \min(k_{max}, \|e\|^2 + k_{off})\mathbf{I} + \mathbf{W}_{adj} + \mathbf{W}_\tau \quad (150)$$

return $\mathbf{W}_{reg} \in \mathbb{R}^{dim(\mathbf{q}) \times dim(\mathbf{q})}$

:regular-vector [method]

$$\mathbf{v}_{reg} := \mathbf{v}_{adj} + \mathbf{v}_\tau \quad (151)$$

return $\mathbf{v}_{reg} \in \mathbb{R}^{dim(\mathbf{q})}$

:update-viewer [method]

Update viewer.

:print-status [method]

Print status.

:play-animation *key* (*robot-env*) [method]

(*loop?* *t*)

(*visualize-callback-func*)

Play motion.

:generate-robot-state-list *key* (*robot-env*) [method]

(*joint-name-list* (*send-all* (*send robot-env :robot :joint-list*) *:name*))

(*root-link-name* (*send* (*car* (*send robot-env :robot :links*)) *:name*))

(*step-time* 0.004)

(*divide-num* 100)

(*limb-list* (*list :rleg :lleg :rarm :larm*))

Generate and return robot state list.

3 逐次二次計画法

sqp-optimization [class]

:super **propertied-object**

:slots (*_config-task* instance of configuration-task)

(*_qp_retval* buffer for QP return value)

(*_qp-status* buffer for QP status)

(*_qp-int-status* QP status)

(*_task-value* buffer for task value $e(\mathbf{q})$)

(*_task-jacobian* buffer for task jacobian $\frac{\partial e}{\partial \mathbf{q}}$)

(*_dim-config-buf-matrix* matrix buffer)

(*_convergence-check-func* function to check convergence)

(*_failure-callback-func* callback function of failure)
 (*_pre-process-func* pre-process function)
 (*_post-process-func* post-process function)
 (*_i* buffer for iteration count)
 (*_status* status of sqp optimization)
 (*_no-visualize?* whether to suppress visualization)
 (*_no-print?* whether to suppress print)

逐次二次計画法のクラス .

instant-configuration-task クラスや trajectory-configuration-task クラスの instance (以降, configuration-task と呼ぶ) が与えられた時に, configuration-task のタスク関数ノルム二乗 $\|e(q)\|^2$ を最小にするコンフィギュレーション q を反復計算により求める .

:init *ℰkey* (*config-task*) [method]
 (*convergence-check-func*)
 (*failure-callback-func*)
 (*pre-process-func*)
 (*post-process-func*)
 (*no-visualize?*)
 (*no-print?*)
 Initialize instance

:config-task [method]
 Return configuration-task instance

:optimize *ℰkey* (*loop-num* 100) [method]
 (*loop-num-min*)
 (*update-viewer-interval* 1)
 (*print-status-interval* 10)

Optimize

In each iteration, do following:

1. check convergence
2. call pre-process function
3. print status
4. solve QP and update configuration
5. call post-process function

Solve following QP:

$$\min_{\Delta \mathbf{q}^{(k)}} \frac{1}{2} \Delta \mathbf{q}^{(k)T} \mathbf{W} \Delta \mathbf{q}^{(k)} + \mathbf{v}^T \Delta \mathbf{q}^{(k)} \quad (152)$$

$$\text{s.t. } \mathbf{A} \Delta \mathbf{q}^{(k)} = \mathbf{b} \quad (153)$$

$$\mathbf{C} \Delta \mathbf{q}^{(k)} \geq \mathbf{d} \quad (154)$$

$$\text{where } \mathbf{W} = \left(\frac{\partial \mathbf{e}(\mathbf{q}^{(k)})}{\partial \mathbf{q}^{(k)}} \right)^T \left(\frac{\partial \mathbf{e}(\mathbf{q}^{(k)})}{\partial \mathbf{q}^{(k)}} \right) + \mathbf{W}_{reg} \quad (155)$$

$$\mathbf{v} = \left(\frac{\partial \mathbf{e}(\mathbf{q}^{(k)})}{\partial \mathbf{q}^{(k)}} \right)^T \mathbf{e}(\mathbf{q}^{(k)}) + \mathbf{v}_{reg} \quad (156)$$

and update configuration:

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \Delta \mathbf{q}^{(k)*} \quad (157)$$

:iteration [method]

Return iteration index.

:status [method]

Return status of sqp optimization.

4 動作生成の拡張

4.1 マニピュレーションの動作生成

robot-object-environment [class]

```
:super    robot-environment
:slots    (_obj  $\mathcal{O}$ )
           (_obj-with-root-virtual  $\hat{\mathcal{O}}$ )
```

ロボットと物体とロボット・環境間の接触のクラス .

以下を合わせた関節・リンク構造に関するメソッドが定義されている .

1. 浮遊ルートリンクのための仮想関節付きのロボットの関節
2. 物体位置姿勢を表す仮想関節
3. 接触位置を定める仮想関節

関節・リンク構造を定めるために , 初期化時に以下を与える

robot \mathcal{R} ロボット (cascaded-link クラスのインスタンス) .

object \mathcal{O} 物体 (cascaded-link クラスのインスタンス) . 関節をもたないことを前提とする .

contact-list $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{N_C}\}$ 接触 (2d-planar-contact クラスなどのインスタンス) のリスト .

ロボット R に , 浮遊ルートリンクの変位に対応する仮想関節を付加した仮想関節付きロボット $\hat{\mathcal{R}}$ を内部で保持する . 同様に , 物体 O に , 物体の変位に対応する仮想関節を付加した仮想関節付き物体 $\hat{\mathcal{O}}$ を内部で保持する .

:init $\mathcal{E}key$ (*robot*) [method]

(*object*)

(*contact-list*)

(*root-virtual-mode* :6dof)

(*root-virtual-joint-class-list*)

(*root-virtual-joint-axis-list*)

Initialize instance

:object $\mathcal{E}rest$ *args* [method]

return \mathcal{O}

:object-with-root-virtual $\mathcal{E}_{rest} \text{ args}$

[method]

return $\hat{\mathcal{O}}$

instant-manipulation-configuration-task

[class]

:super **instant-configuration-task**

:slots (**_robot-obj-env** robot-object-environment instance)

(**_wrench-obj-vector** $\hat{\mathbf{w}}_{obj}$ [N] [Nm])

(**_num-contact-obj** $N_{cnt-obj} := |\mathcal{T}^{cnt-trg-obj}|$)

(**_num-act-react** $N_{act-react} := |\mathcal{P}^{act-react}|$)

(**_dim-wrench-obj** $\dim(\hat{\mathbf{w}}_{obj}) = 6N_{cnt-obj}$)

(**_contact-target-coords-obj-list** $\mathcal{T}^{cnt-trg-obj}$)

(**_contact-constraint-obj-list** list of contact-constraint instance for object)

(**_act-react-pair-list** $\mathcal{P}^{act-react}$)

マニピュレーションにける瞬時コンフィギュレーション $q^{(l)}$ と瞬時タスク関数 $e^{(l)}(q^{(l)})$ のクラス．マニピュレーション対象の物体の瞬時コンフィギュレーションや瞬時タスク関数を含む．

このクラスの説明で用いる全ての変数は，時間ステップ l を表す添字をつけて $x^{(l)}$ と表されるべきだが，このクラス内の説明では省略して x と表す．また，以降では，説明文やメソッド名で，“瞬時”や“instant”を省略する．

コンフィギュレーション q の取得・更新，タスク関数 $e(q)$ の取得，タスク関数のヤコビ行列 $\frac{\partial e(q)}{\partial q}$ の取得，コンフィギュレーションの等式・不等式制約 A, b, C, d の取得のためのメソッドが定義されている．

コンフィギュレーション・タスク関数を定めるために，instant-configuration-task の設定に加えて，初期化時に以下を与える

- ロボット・物体・環境

robot-object-environment ロボット・物体・環境を表す robot-object-environment クラスのインスタンス

- 物体の接触拘束

contact-target-coords-obj-list $\mathcal{T}^{cnt-trg-obj}$ 物体の接触目標位置姿勢リスト

contact-constraint-obj-list 物体の接触レンチ制約リスト

- 作用・反作用の拘束

act-react-pair-list $\mathcal{P}^{act-react}$ 作用・反作用の関係にあるロボット・物体の接触目標位置姿勢ペアのリスト

コンフィギュレーション q は以下から構成される．

$\theta \in \mathbb{R}^{N_{var-joint}}$ 時変関節角度 [rad] [m]

$\hat{\mathbf{w}} \in \mathbb{R}^{6N_{cnt}}$ ロボットの接触レンチ [N] [Nm]

$\hat{\mathbf{w}}_{obj} \in \mathbb{R}^{6N_{cnt-obj}}$ 物体の接触レンチ [N] [Nm]

$\tau \in \mathbb{R}^{N_{drive-joint}}$ 関節駆動トルク [Nm] [N]

$\phi \in \mathbb{R}^{N_{invar-joint}}$ 時不変関節角度 [rad] [m]

$$\text{return } \mathbf{q}_{var} := \begin{pmatrix} \boldsymbol{\theta} \\ \hat{\mathbf{w}} \\ \hat{\mathbf{w}}_{obj} \\ \boldsymbol{\tau} \end{pmatrix}$$

:config-vector

[method]

$$\text{return } \mathbf{q} := \begin{pmatrix} \mathbf{q}_{var} \\ \mathbf{q}_{invar} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\theta} \\ \hat{\mathbf{w}} \\ \hat{\mathbf{w}}_{obj} \\ \boldsymbol{\tau} \\ \boldsymbol{\phi} \end{pmatrix}$$

:set-wrench-obj *wrench-obj-new* *ℰkey* (*relative?* *nil*)

[method]

Set $\hat{\mathbf{w}}_{obj}$.

:set-variant-config *variant-config-new* *ℰkey* (*relative?* *nil*)

[method]

(*apply-to-robot?* *t*)

Set \mathbf{q}_{var} .

:contact-target-coords-obj-list

[method]

$$T_m^{cnt-trg-obj} = \{\mathbf{p}_m^{cnt-trg-obj}, \mathbf{R}_m^{cnt-trg-obj}\} \quad (m = 1, 2, \dots, N_{cnt-obj}) \quad (164)$$

$$\text{return } \mathcal{T}^{cnt-trg-obj} := \{T_1^{cnt-trg-obj}, T_2^{cnt-trg-obj}, \dots, T_{N_{cnt-obj}}^{cnt-trg-obj}\}$$

:contact-constraint-obj-list

[method]

return list of contact-constraint instance for object

:wrench-obj-list

[method]

$$\text{return } \{\mathbf{w}_{obj,1}, \mathbf{w}_{obj,2}, \dots, \mathbf{w}_{obj,N_{obj}}\}$$

:force-obj-list

[method]

$$\text{return } \{\mathbf{f}_{obj,1}, \mathbf{f}_{obj,2}, \dots, \mathbf{f}_{obj,N_{cnt-obj}}\}$$

:moment-obj-list

[method]

$$\text{return } \{\mathbf{n}_{obj,1}, \mathbf{n}_{obj,2}, \dots, \mathbf{n}_{obj,N_{cnt-obj}}\}$$

:mg-obj-vec

[method]

$$\text{return } m_{obj} \mathbf{g}$$

:cog-obj *ℰkey* (*update?* *t*)

[method]

$$\text{return } \mathbf{p}_{Gobj}(\mathbf{q})$$

:eom-trans-obj-task-value *ℰkey* (*update?* *t*)

[method]

$$\mathbf{e}^{eom-trans-obj}(\mathbf{q}) = \mathbf{e}^{eom-trans-obj}(\hat{\mathbf{w}}_{obj}) \quad (165)$$

$$= \sum_{m=1}^{N_{cnt-obj}} \mathbf{f}_{obj,m} - m_{obj} \mathbf{g} \quad (166)$$

$$\text{return } \mathbf{e}^{eom-trans-obj}(\mathbf{q}) \in \mathbb{R}^3$$

:eom-rot-obj-task-value $\mathcal{E}key$ (*update?* t) [method]

$$\mathbf{e}^{eom-rot-obj}(\mathbf{q}) = \mathbf{e}^{eom-rot-obj}(\boldsymbol{\theta}, \hat{\mathbf{w}}_{obj}, \phi) \quad (167)$$

$$= \sum_{m=1}^{N_{cnt-obj}} \{ (\mathbf{p}_m^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) - \mathbf{p}_{Gobj}(\boldsymbol{\theta}, \phi)) \times \mathbf{f}_{obj,m} + \mathbf{n}_{obj,m} \} \quad (168)$$

return $\mathbf{e}^{eom-rot-obj}(\mathbf{q}) \in \mathbb{R}^3$

:task-value $\mathcal{E}key$ (*update?* t) [method]

$$\text{return } \mathbf{e}(\mathbf{q}) := \begin{pmatrix} \mathbf{e}^{kin}(\mathbf{q}) \\ \mathbf{e}^{eom-trans}(\mathbf{q}) \\ \mathbf{e}^{eom-rot}(\mathbf{q}) \\ \mathbf{e}^{eom-trans-obj}(\mathbf{q}) \\ \mathbf{e}^{eom-rot-obj}(\mathbf{q}) \\ \mathbf{e}^{trq}(\mathbf{q}) \\ \mathbf{e}^{posture}(\mathbf{q}) \end{pmatrix} = \begin{pmatrix} \mathbf{e}^{kin}(\boldsymbol{\theta}, \phi) \\ \mathbf{e}^{eom-trans}(\hat{\mathbf{w}}) \\ \mathbf{e}^{eom-rot}(\boldsymbol{\theta}, \hat{\mathbf{w}}, \phi) \\ \mathbf{e}^{eom-trans-obj}(\hat{\mathbf{w}}_{obj}) \\ \mathbf{e}^{eom-rot-obj}(\boldsymbol{\theta}, \hat{\mathbf{w}}_{obj}, \phi) \\ \mathbf{e}^{trq}(\boldsymbol{\theta}, \hat{\mathbf{w}}, \tau, \phi) \\ \mathbf{e}^{posture}(\boldsymbol{\theta}) \end{pmatrix}$$

:eom-trans-obj-task-jacobian-with-wrench-obj [method]

$$\frac{\partial \mathbf{e}^{eom-trans-obj}}{\partial \hat{\mathbf{w}}_{obj}} = \begin{pmatrix} \frac{\partial \mathbf{e}^{eom-trans-obj}}{\partial \mathbf{f}_{obj,1}} & \frac{\partial \mathbf{e}^{eom-trans-obj}}{\partial \mathbf{n}_{obj,1}} & \cdots & \frac{\partial \mathbf{e}^{eom-trans-obj}}{\partial \mathbf{f}_{obj,N_{cnt-obj}}} & \frac{\partial \mathbf{e}^{eom-trans-obj}}{\partial \mathbf{n}_{obj,N_{cnt-obj}}} \end{pmatrix} \quad (169)$$

$$= \begin{pmatrix} \mathbf{I}_3 & \mathbf{O}_3 & \cdots & \mathbf{I}_3 & \mathbf{O}_3 \end{pmatrix} \quad (170)$$

return $\frac{\partial \mathbf{e}^{eom-trans-obj}}{\partial \hat{\mathbf{w}}_{obj}} \in \mathbb{R}^{3 \times 6N_{cnt-obj}}$

:eom-rot-obj-task-jacobian-with-theta [method]

$$\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \boldsymbol{\theta}} = \sum_{m=1}^{N_{cnt-obj}} \left\{ -[\mathbf{f}_{obj,m} \times] \left(\mathbf{J}_{\theta,m}^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{Gobj\theta}(\boldsymbol{\theta}, \phi) \right) \right\} \quad (171)$$

$$= \left[\left(\sum_{m=1}^{N_{cnt-obj}} \mathbf{f}_{obj,m} \right) \times \right] \mathbf{J}_{Gobj\theta}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{cnt-obj}} [\mathbf{f}_{obj,m} \times] \mathbf{J}_{\theta,m}^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) \quad (172)$$

$\sum_{m=1}^{N_{cnt-obj}} \mathbf{f}_{obj,m} = m_{obj} \mathbf{g}$ つまり, eom-trans-obj-task が成立すると仮定すると次式が成り立つ .

$$\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \boldsymbol{\theta}} = [m_{obj} \mathbf{g} \times] \mathbf{J}_{Gobj\theta}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{cnt-obj}} [\mathbf{f}_{obj,m} \times] \mathbf{J}_{\theta,m}^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) \quad (173)$$

return $\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{3 \times N_{var-joint}}$

:eom-rot-obj-task-jacobian-with-wrench-obj [method]

$$\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \hat{\mathbf{w}}_{obj}} = \begin{pmatrix} \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \mathbf{f}_{obj,1}} & \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \mathbf{n}_{obj,1}} & \cdots & \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \mathbf{f}_{obj,N_{cnt-obj}}} & \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \mathbf{n}_{obj,N_{cnt-obj}}} \end{pmatrix} \quad (174)$$

$$\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \mathbf{f}_{obj,m}} = [(\mathbf{p}_m^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) - \mathbf{p}_{Gobj}(\boldsymbol{\theta}, \phi)) \times] \quad (m = 1, 2, \dots, N_{cnt-obj}) \quad (175)$$

$$\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \mathbf{n}_{obj,m}} = \mathbf{I}_3 \quad (m = 1, 2, \dots, N_{cnt-obj}) \quad (176)$$

return $\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \hat{\mathbf{w}}_{obj}} \in \mathbb{R}^{3 \times 6N_{cnt-obj}}$

:eom-rot-obj-task-jacobian-with-phi

[method]

$$\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \phi} = \sum_{m=1}^{N_{cnt-obj}} \left\{ -[\mathbf{f}_{obj,m} \times] \left(\mathbf{J}_{\phi,m}^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) - \mathbf{J}_{Gobj\phi}(\boldsymbol{\theta}, \phi) \right) \right\} \quad (177)$$

$$= \left[\left(\sum_{m=1}^{N_{cnt-obj}} \mathbf{f}_{obj,m} \right) \times \right] \mathbf{J}_{Gobj\phi}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{cnt-obj}} [\mathbf{f}_{obj,m} \times] \mathbf{J}_{\phi,m}^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) \quad (178)$$

$\sum_{m=1}^{N_{cnt-obj}} \mathbf{f}_{obj,m} = m_{obj} \mathbf{g}$ つまり, eom-trans-obj-task が成立すると仮定すると次式が成り立つ .

$$\frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \phi} = [m_{obj} \mathbf{g} \times] \mathbf{J}_{Gobj\phi}(\boldsymbol{\theta}, \phi) - \sum_{m=1}^{N_{cnt-obj}} [\mathbf{f}_{obj,m} \times] \mathbf{J}_{\phi,m}^{cnt-trg-obj}(\boldsymbol{\theta}, \phi) \quad (179)$$

$$\text{return } \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \phi} \in \mathbb{R}^{3 \times N_{invar-joint}}$$

:variant-task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} = \begin{matrix} & N_{var-joint} & 6N_{cnt} & 6N_{cnt-obj} & N_{drive-joint} \\ \begin{matrix} 6N_{kin} \\ 3 \\ 3 \\ 3 \\ 3 \\ N_{drive-joint} \\ N_{posture-joint} \end{matrix} & \begin{pmatrix} \frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\theta}} & & & \\ & \frac{\partial \mathbf{e}^{eom-trans}}{\partial \hat{\mathbf{w}}} & & \\ & \frac{\partial \mathbf{e}^{eom-rot}}{\partial \hat{\mathbf{w}}} & & \\ & & \frac{\partial \mathbf{e}^{eom-trans-obj}}{\partial \hat{\mathbf{w}}_{obj}} & \\ & & \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \hat{\mathbf{w}}_{obj}} & \\ & \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{e}^{trq}}{\partial \hat{\mathbf{w}}} & \\ & \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\theta}} & & \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\tau}} \\ & \frac{\partial \mathbf{e}^{posture}}{\partial \boldsymbol{\theta}} & & \end{pmatrix} \end{matrix} \quad (180)$$

$$\text{return } \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{var}} \in \mathbb{R}^{(6N_{kin}+3+3+3+3+N_{drive-joint}+N_{posture-joint}) \times (N_{var-joint}+6N_{cnt}+6N_{cnt-obj}+N_{drive-joint})}$$

:invariant-task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} = \begin{matrix} & N_{invar-joint} \\ \begin{matrix} 6N_{kin} \\ 3 \\ 3 \\ 3 \\ 3 \\ N_{drive-joint} \\ N_{posture-joint} \end{matrix} & \begin{pmatrix} \frac{\partial \mathbf{e}^{kin}}{\partial \phi} \\ \\ \frac{\partial \mathbf{e}^{eom-rot}}{\partial \phi} \\ \\ \frac{\partial \mathbf{e}^{eom-rot-obj}}{\partial \phi} \\ \frac{\partial \mathbf{e}^{trq}}{\partial \phi} \end{pmatrix} \end{matrix} \quad (181)$$

$$\text{return } \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{invar}} \in \mathbb{R}^{(6N_{kin}+3+3+3+3+N_{drive-joint}+N_{posture-joint}) \times N_{invar-joint}}$$

:task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}} = \begin{pmatrix} \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{\text{var}}} & \frac{\partial \mathbf{e}}{\partial \mathbf{q}_{\text{invar}}} \end{pmatrix} \quad (182)$$

$$= \begin{matrix} 6N_{kin} \\ 3 \\ 3 \\ 3 \\ 3 \\ N_{drive-joint} \\ N_{posture-joint} \end{matrix} \begin{pmatrix} N_{var-joint} & 6N_{cnt} & 6N_{cnt-obj} & N_{drive-joint} & N_{invar-joint} \\ \frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\theta}} & & & & \frac{\partial \mathbf{e}^{kin}}{\partial \boldsymbol{\phi}} \\ & \frac{\partial \mathbf{e}^{com-trans}}{\partial \hat{\mathbf{w}}} & & & \\ \frac{\partial \mathbf{e}^{com-rot}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{e}^{com-rot}}{\partial \hat{\mathbf{w}}} & & & \frac{\partial \mathbf{e}^{com-rot}}{\partial \boldsymbol{\phi}} \\ & & \frac{\partial \mathbf{e}^{com-trans-obj}}{\partial \hat{\mathbf{w}}_{obj}} & & \\ \frac{\partial \mathbf{e}^{com-rot-obj}}{\partial \boldsymbol{\theta}} & & \frac{\partial \mathbf{e}^{com-rot-obj}}{\partial \hat{\mathbf{w}}_{obj}} & & \frac{\partial \mathbf{e}^{com-rot-obj}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{e}^{trq}}{\partial \hat{\mathbf{w}}} & & \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\tau}} & \frac{\partial \mathbf{e}^{trq}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \mathbf{e}^{posture}}{\partial \boldsymbol{\theta}} & & & & \end{pmatrix} \quad (183)$$

$$\text{return } \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \in \mathbb{R}^{(6N_{kin}+3+3+3+3+N_{drive-joint}+N_{posture-joint}) \times (N_{var-joint}+6N_{cnt}+6N_{cnt-obj}+N_{drive-joint}+N_{invar-joint})}$$

:wrench-obj-inequality-constraint-matrix *ℰkey (update? t)* [method]

物体の接触レンチ $\mathbf{w}_{obj} \in \mathbb{R}^6$ が満たすべき制約（非負制約，摩擦制約，圧力中心制約）が次式のように表されるとする．

$$\mathbf{C}_{w_{obj}} \mathbf{w}_{obj} \geq \mathbf{d}_{w_{obj}} \quad (184)$$

$N_{cnt-obj}$ 箇所の接触部位の接触レンチを並べたベクトル $\hat{\mathbf{w}}_{obj}$ の不等式制約は次式で表される．

$$\mathbf{C}_{w_{obj},m} (\mathbf{w}_{obj,m} + \Delta \mathbf{w}_{obj,m}) \geq \mathbf{d}_{w_{obj},m} \quad (m = 1, 2, \dots, N_{cnt-obj}) \quad (185)$$

$$\Leftrightarrow \mathbf{C}_{w_{obj},m} \Delta \mathbf{w}_{obj,m} \geq \mathbf{d}_{w_{obj},m} - \mathbf{C}_{w_{obj},m} \mathbf{w}_{obj,m} \quad (m = 1, 2, \dots, N_{cnt-obj}) \quad (186)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{C}_{w_{obj},1} & & & \\ & \mathbf{C}_{w_{obj},2} & & \\ & & \ddots & \\ & & & \mathbf{C}_{w_{obj},N_{cnt-obj}} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{w}_{obj,1} \\ \Delta \mathbf{w}_{obj,2} \\ \vdots \\ \Delta \mathbf{w}_{obj,N_{cnt-obj}} \end{pmatrix} \geq \begin{pmatrix} \mathbf{d}_{w_{obj},1} - \mathbf{C}_{w_{obj},1} \mathbf{w}_{obj,1} \\ \mathbf{d}_{w_{obj},2} - \mathbf{C}_{w_{obj},2} \mathbf{w}_{obj,2} \\ \vdots \\ \mathbf{d}_{w_{obj},N_{cnt-obj}} - \mathbf{C}_{w_{obj},N_{cnt-obj}} \mathbf{w}_{obj,N_{cnt-obj}} \end{pmatrix} \quad (187)$$

$$\Leftrightarrow \mathbf{C}_{\hat{\mathbf{w}}_{obj}} \Delta \hat{\mathbf{w}}_{obj} \geq \mathbf{d}_{\hat{\mathbf{w}}_{obj}} \quad (188)$$

$$\text{return } \mathbf{C}_{\hat{\mathbf{w}}_{obj}} := \begin{pmatrix} \mathbf{C}_{w_{obj},1} & & & \\ & \mathbf{C}_{w_{obj},2} & & \\ & & \ddots & \\ & & & \mathbf{C}_{w_{obj},N_{cnt-obj}} \end{pmatrix} \in \mathbb{R}^{N_{wrench-obj-ineq} \times 6N_{cnt-obj}}$$

:wrench-obj-inequality-constraint-vector *ℰkey (update? t)* [method]

$$\text{return } \mathbf{d}_{\hat{\mathbf{w}}_{obj}} := \begin{pmatrix} \mathbf{d}_{w_{obj},1} - \mathbf{C}_{w_{obj},1} \mathbf{w}_{obj,1} \\ \mathbf{d}_{w_{obj},2} - \mathbf{C}_{w_{obj},2} \mathbf{w}_{obj,2} \\ \vdots \\ \mathbf{d}_{w_{obj},N_{cnt-obj}} - \mathbf{C}_{w_{obj},N_{cnt-obj}} \mathbf{w}_{obj,N_{cnt-obj}} \end{pmatrix} \in \mathbb{R}^{N_{wrench-obj-ineq}}$$

:variant-config-inequality-constraint-matrix *ℰkey (update? nil)* [method]

$$\begin{cases} \mathbf{C}_\theta \Delta \boldsymbol{\theta} \geq \mathbf{d}_\theta \\ \mathbf{C}_{\hat{\mathbf{w}}} \Delta \hat{\mathbf{w}} \geq \mathbf{d}_{\hat{\mathbf{w}}} \\ \mathbf{C}_{\hat{\mathbf{w}}_{obj}} \Delta \hat{\mathbf{w}}_{obj} \geq \mathbf{d}_{\hat{\mathbf{w}}_{obj}} \\ \mathbf{C}_\tau \Delta \boldsymbol{\tau} \geq \mathbf{d}_\tau \end{cases} \quad (189)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{C}_\theta & & & \\ & \mathbf{C}_{\hat{\mathbf{w}}} & & \\ & & \mathbf{C}_{\hat{\mathbf{w}}_{obj}} & \\ & & & \mathbf{C}_\tau \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{\theta} \\ \Delta \hat{\mathbf{w}} \\ \Delta \hat{\mathbf{w}}_{obj} \\ \Delta \boldsymbol{\tau} \end{pmatrix} \geq \begin{pmatrix} \mathbf{d}_\theta \\ \mathbf{d}_{\hat{\mathbf{w}}} \\ \mathbf{d}_{\hat{\mathbf{w}}_{obj}} \\ \mathbf{d}_\tau \end{pmatrix} \quad (190)$$

$$\Leftrightarrow \mathbf{C}_{var} \Delta \mathbf{q}_{var} \geq \mathbf{d}_{var} \quad (191)$$

$$\text{return } \mathbf{C}_{var} := \begin{pmatrix} \mathbf{C}_\theta & & & \\ & \mathbf{C}_{\hat{\mathbf{w}}} & & \\ & & \mathbf{C}_{\hat{\mathbf{w}}_{obj}} & \\ & & & \mathbf{C}_\tau \end{pmatrix} \in \mathbb{R}^{N_{var-ineq} \times \dim(\mathbf{q}_{var})}$$

:variant-config-inequality-constraint-vector *ℓkey (update? t)* [method]

$$\text{return } \mathbf{d}_{var} := \begin{pmatrix} \mathbf{d}_\theta \\ \mathbf{d}_{\hat{\mathbf{w}}} \\ \mathbf{d}_{\hat{\mathbf{w}}_{obj}} \\ \mathbf{d}_\tau \end{pmatrix} \in \mathbb{R}^{N_{var-ineq}}$$

:act-react-equality-constraint-matrix *ℓkey (update? nil)* [method]

ロボット・物体間の接触レンチに関する作用・反作用の法則は次式のように表される．

$$\hat{\mathbf{w}}_{i(m)} + \hat{\mathbf{w}}_{obj,j(m)} = \mathbf{0} \quad (m = 1, 2, \dots, N_{act-react}) \quad (192)$$

$$\Leftrightarrow \mathbf{A}_{act-react,robot,m} \hat{\mathbf{w}} + \mathbf{A}_{act-react,obj,m} \hat{\mathbf{w}}_{obj} = \mathbf{0} \quad (m = 1, 2, \dots, N_{act-react}) \quad (193)$$

$$\text{where } \mathbf{A}_{act-react,robot,m} = \begin{pmatrix} \mathbf{O}_6 & \mathbf{O}_6 & \cdots & \mathbf{I}_6 & \cdots & \mathbf{O}_6 & \mathbf{O}_6 \end{pmatrix} \in \mathbb{R}^{6 \times 6N_{cnt}} \quad (194)$$

$$\mathbf{A}_{act-react,obj,m} = \begin{pmatrix} \mathbf{O}_6 & \mathbf{O}_6 & \cdots & \mathbf{I}_6 & \cdots & \mathbf{O}_6 & \mathbf{O}_6 \end{pmatrix} \in \mathbb{R}^{6 \times 6N_{cnt-obj}} \quad (195)$$

$$\Leftrightarrow \mathbf{A}_{act-react,robot} \hat{\mathbf{w}} + \mathbf{A}_{act-react,obj} \hat{\mathbf{w}}_{obj} = \mathbf{0} \quad (196)$$

$$\text{where } \mathbf{A}_{act-react,robot} = \begin{pmatrix} \mathbf{A}_{act-react,robot,1} \\ \vdots \\ \mathbf{A}_{act-react,robot,N_{act-react}} \end{pmatrix} \in \mathbb{R}^{6N_{act-react} \times 6N_{cnt}} \quad (197)$$

$$\mathbf{A}_{act-react,obj} = \begin{pmatrix} \mathbf{A}_{act-react,obj,1} \\ \vdots \\ \mathbf{A}_{act-react,obj,N_{act-react}} \end{pmatrix} \in \mathbb{R}^{6N_{act-react} \times 6N_{cnt-obj}} \quad (198)$$

$$\Leftrightarrow \mathbf{A}_{act-react} \begin{pmatrix} \hat{\mathbf{w}} \\ \hat{\mathbf{w}}_{obj} \end{pmatrix} = \mathbf{0} \in \mathbb{R}^{6N_{act-react}} \quad (199)$$

$$\text{where } \mathbf{A}_{act-react} = \begin{pmatrix} \mathbf{A}_{act-react,robot} & \mathbf{A}_{act-react,obj} \end{pmatrix} \in \mathbb{R}^{6N_{act-react} \times (6N_{cnt} + 6N_{cnt-obj})} \quad (200)$$

$$\Leftrightarrow \mathbf{A}_{act-react} \begin{pmatrix} \hat{\mathbf{w}} + \Delta \hat{\mathbf{w}} \\ \hat{\mathbf{w}}_{obj} + \Delta \hat{\mathbf{w}}_{obj} \end{pmatrix} = \mathbf{0} \quad (201)$$

$$\Leftrightarrow \mathbf{A}_{act-react} \begin{pmatrix} \Delta \hat{\mathbf{w}} \\ \Delta \hat{\mathbf{w}}_{obj} \end{pmatrix} = \mathbf{b}_{act-react} \quad (202)$$

$$\text{where } \mathbf{b}_{act-react} = -\mathbf{A}_{act-react} \begin{pmatrix} \hat{\mathbf{w}} \\ \hat{\mathbf{w}}_{obj} \end{pmatrix} \quad (203)$$

$i(m), j(m)$ は作用・反作用の関係にある接触レンチの m 番目の対におけるロボット，物体の接触レンチのインデックスである．

return $\mathbf{A}_{act-react} \in \mathbb{R}^{6N_{act-react} \times (6N_{cnt} + 6N_{cnt-obj})}$

:act-react-equality-constraint-vector $\mathcal{E}key$ (update? t) [method]

return $\mathbf{b}_{act-react} \in \mathbb{R}^{6N_{act-react}}$

:variant-config-equality-constraint-matrix $\mathcal{E}key$ (update? nil) [method]

$$\mathbf{A}_{act-react} \begin{pmatrix} \Delta \hat{\mathbf{w}} \\ \Delta \hat{\mathbf{w}}_{obj} \end{pmatrix} = \mathbf{b}_{act-react} \quad (204)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{O} & \mathbf{A}_{act-react} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \Delta \theta \\ \Delta \hat{\mathbf{w}} \\ \Delta \hat{\mathbf{w}}_{obj} \\ \Delta \tau \end{pmatrix} = \mathbf{b}_{act-react} \quad (205)$$

$$\Leftrightarrow \mathbf{A}_{var} \Delta \mathbf{q}_{var} = \mathbf{b}_{var} \quad (206)$$

return $\mathbf{A}_{var} := \begin{pmatrix} \mathbf{O} & \mathbf{A}_{act-react} & \mathbf{O} \end{pmatrix} \in \mathbb{R}^{6N_{act-react} \times \dim(\mathbf{q}_{var})}$

:variant-config-equality-constraint-vector $\mathcal{E}key$ (*update?* t) [method]

return $\mathbf{b}_{var} := \mathbf{b}_{act-react} \in \mathbb{R}^{6N_{act-react}}$

:invariant-config-equality-constraint-matrix $\mathcal{E}key$ (*update?* nil) [method]

return $\mathbf{A}_{invar} \in \mathbb{R}^{0 \times \dim(\mathbf{q}_{invar})}$ (no equality constraint)

:invariant-config-equality-constraint-vector $\mathcal{E}key$ (*update?* t) [method]

return $\mathbf{b}_{invar} \in \mathbb{R}^0$ (no equality constraint)

:config-equality-constraint-matrix $\mathcal{E}key$ (*update?* nil) [method]

$$\mathbf{A}_{var} \Delta \mathbf{q}_{var} = \mathbf{b}_{var} \quad (207)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{A}_{var} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{q}_{var} \\ \Delta \mathbf{q}_{invar} \end{pmatrix} = \mathbf{b}_{var} \quad (208)$$

$$\Leftrightarrow \mathbf{A} \Delta \mathbf{q} = \mathbf{b} \quad (209)$$

return $\mathbf{A} := \begin{pmatrix} \mathbf{A}_{var} & \mathbf{O} \end{pmatrix} \in \mathbb{R}^{N_{eq} \times \dim(\mathbf{q})}$

:config-equality-constraint-vector $\mathcal{E}key$ (*update?* t) [method]

return $\mathbf{b} := \mathbf{b}_{var} \in \mathbb{R}^{N_{eq}}$

:torque-regular-matrix $\mathcal{E}key$ (*update?* nil) [method]

(*only-variant?* nil)

二次形式の正則化項として次式を考える .

$$F_{tau}(\mathbf{q}) = \left\| \frac{\boldsymbol{\tau}}{\boldsymbol{\tau}_{max}} \right\|^2 \quad (\text{ベクトルの要素ごとの割り算を表す}) \quad (210)$$

$$= \boldsymbol{\tau}^T \bar{\mathbf{W}}_{\tau} \boldsymbol{\tau} \quad (211)$$

ここで ,

$$\bar{\mathbf{W}}_{\tau} := \begin{pmatrix} \frac{1}{\tau_{max,1}^2} & & & \\ & \frac{1}{\tau_{max,2}^2} & & \\ & & \ddots & \\ & & & \frac{1}{\tau_{max,N_{drive-joint}}^2} \end{pmatrix} \in \mathbb{R}^{\dim(\boldsymbol{\tau}) \times \dim(\boldsymbol{\tau})} \quad (212)$$

only-variant? is true:

$$\mathbf{W}_{\tau} := \begin{matrix} & \dim(\boldsymbol{\theta}) & \dim(\hat{\mathbf{w}}) & \dim(\hat{\mathbf{w}}_{obj}) & \dim(\boldsymbol{\tau}) \\ \begin{matrix} \dim(\boldsymbol{\theta}) \\ \dim(\hat{\mathbf{w}}) \\ \dim(\hat{\mathbf{w}}_{obj}) \\ \dim(\boldsymbol{\tau}) \end{matrix} & \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \bar{\mathbf{W}}_{\tau} \end{pmatrix} & \in \mathbb{R}^{\dim(\mathbf{q}_{var}) \times \dim(\mathbf{q}_{var})} \end{matrix} \quad (213)$$

otherwise:

$$\mathbf{W}_{\tau} := \begin{matrix} & \dim(\boldsymbol{\theta}) & \dim(\hat{\mathbf{w}}) & \dim(\hat{\mathbf{w}}_{obj}) & \dim(\boldsymbol{\tau}) & \dim(\boldsymbol{\phi}) \\ \begin{matrix} \dim(\boldsymbol{\theta}) \\ \dim(\hat{\mathbf{w}}) \\ \dim(\hat{\mathbf{w}}_{obj}) \\ \dim(\boldsymbol{\tau}) \\ \dim(\boldsymbol{\phi}) \end{matrix} & \begin{pmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \bar{\mathbf{W}}_{\tau} \\ & & & & \end{pmatrix} & \in \mathbb{R}^{\dim(\mathbf{q}) \times \dim(\mathbf{q})} \end{matrix} \quad (214)$$

$$\text{return } \mathbf{W}_\tau$$

```

:torque-regular-vector key (update? t) [method]
                        (only-variant? nil)

```

$$\bar{\mathbf{v}}_\tau := \bar{\mathbf{W}}_\tau \boldsymbol{\tau} \quad (215)$$

$$= \begin{pmatrix} \frac{\tau_1}{\tau_{\max,1}^2} \\ \frac{\tau_2}{\tau_{\max,2}^2} \\ \vdots \\ \frac{\tau_{\dim(\mathbf{T})}}{\tau_{\max,\dim(\mathbf{T})}^2} \end{pmatrix} \in \mathbb{R}^{\dim(\mathbf{T})} \quad (216)$$

only-variant? is true:

$$\mathbf{v}_\tau := \begin{pmatrix} 1 \\ \dim(\boldsymbol{\theta}) \\ \dim(\hat{\mathbf{w}}) \\ \dim(\hat{\mathbf{w}}_{obj}) \\ \dim(\boldsymbol{\tau}) \\ \bar{\mathbf{v}}_\tau \end{pmatrix} \in \mathbb{R}^{\dim(\mathbf{q}_{var})} \quad (217)$$

otherwise:

$$\mathbf{v}_\tau := \begin{pmatrix} 1 \\ \dim(\boldsymbol{\theta}) \\ \dim(\hat{\mathbf{w}}) \\ \dim(\hat{\mathbf{w}}_{obj}) \\ \dim(\boldsymbol{\tau}) \\ \dim(\phi) \end{pmatrix} \bar{\mathbf{v}}_\tau \in \mathbb{R}^{\dim(\mathbf{q})} \quad (218)$$

$$\text{return } \mathbf{v}_\tau$$

```
:collision-inequality-constraint-matrix key (update? nil) [method]
```

$$\mathbf{C}_{col} := N_{col} \begin{pmatrix} \dim(\boldsymbol{\theta}) & \dim(\hat{\mathbf{w}}) & \dim(\hat{\mathbf{w}}_{obj}) & \dim(\boldsymbol{\tau}) & \dim(\boldsymbol{\phi}) \\ \mathbf{C}_{col,\theta} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{C}_{col,\phi} \end{pmatrix} \quad (219)$$

```

return  $\mathbf{C}_{col} \in \mathbb{R}^{N_{col} \times \dim(\mathbf{q})}$ 

```

| | |
|-----------------------|----------|
| :update-viewer | [method] |
| Update viewer. | |

| | |
|----------------------|----------|
| :print-status | [method] |
| Print status. | |

4.2 B スプラインを用いた関節軌道生成

4.2.1 B スプラインを用いた関節軌道生成の基礎

一般の B スプライン基底関数の定義

B スプライン基底関数は以下で定義される .

$$b_{i,0}(t) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (220)$$

$$b_{i,n}(t) \stackrel{\text{def}}{=} \frac{t - t_i}{t_{i+n} - t_i} b_{i,n-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} b_{i+1,n-1}(t) \quad (221)$$

t_i はノットと呼ばれる .

使用区間を指定してノットを一様とする場合の B スプライン基底関数

t_s, t_f を B スプラインの使用区間の初期 , 終端時刻とする .

$n < m$ とする .

$$t_n = t_s \quad (222)$$

$$t_m = t_f \quad (223)$$

とする . t_i ($0 \leq i \leq n+m$) が等間隔に並ぶとすると ,

$$t_i = \frac{i-n}{m-n}(t_f - t_s) + t_s \quad (224)$$

$$= hi + \frac{mt_s - nt_f}{m-n} \quad (225)$$

ただし ,

$$h \stackrel{\text{def}}{=} \frac{t_f - t_s}{m-n} \quad (226)$$

式 (225) を式 (220) , 式 (221) に代入すると , B スプライン基底関数は次式で得られる .

$$b_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (227)$$

$$b_{i,n}(t) = \frac{(t - t_i)b_{i,n-1}(t) + (t_{i+n+1} - t)b_{i+1,n-1}(t)}{nh} \quad (228)$$

以降では , n を B スプラインの次数 , m を制御点の個数と呼ぶ .

B スプラインの凸包性

式 (227) , 式 (228) で定義される B スプライン基底関数 $b_{i,n}(t)$ は次式のように凸包性を持つ .

$$\sum_{i=0}^{m-1} b_{i,n}(t) = 1 \quad (t_s \leq t \leq t_f) \quad (229)$$

$$0 \leq b_{i,n}(t) \leq 1 \quad (i = 0, 1, \dots, m-1, t_s \leq t \leq t_f) \quad (230)$$

B スプラインの微分

B スプライン基底関数の微分に関して次式が成り立つ⁸ .

$$\dot{\mathbf{b}}_n(t) = \frac{d\mathbf{b}_n(t)}{dt} = \mathbf{D}\mathbf{b}_{n-1}(t) \quad (231)$$

ただし ,

$$\mathbf{b}_n(t) \stackrel{\text{def}}{=} \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{m-1,n}(t) \end{pmatrix} \in \mathbb{R}^m \quad (232)$$

$$\mathbf{D} \stackrel{\text{def}}{=} \frac{1}{h} \begin{pmatrix} 1 & -1 & & & \mathbf{O} \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -1 \\ \mathbf{O} & & & & 1 \end{pmatrix} \in \mathbb{R}^{m \times m} \quad (233)$$

したがって , k 階微分に関して次式が成り立つ .

$$\mathbf{b}_n^{(k)}(t) = \frac{d^{(k)}\mathbf{b}_n(t)}{dt^{(k)}} = \mathbf{D}^k \mathbf{b}_{n-k}(t) \quad (234)$$

B スプラインによる関節角軌道の表現

j 番目の関節角軌道 $\theta_j(t)$ を次式で表す .

$$\theta_j(t) \stackrel{\text{def}}{=} \sum_{i=0}^{m-1} p_{j,i} b_{i,n}(t) = \mathbf{p}_j^T \mathbf{b}_n(t) \in \mathbb{R} \quad (t_s \leq t \leq t_f) \quad (235)$$

ただし ,

$$\mathbf{p}_j = \begin{pmatrix} p_{j,0} \\ p_{j,1} \\ \vdots \\ p_{j,m-1} \end{pmatrix} \in \mathbb{R}^m, \quad \mathbf{b}_n(t) = \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{m-1,n}(t) \end{pmatrix} \in \mathbb{R}^m \quad (236)$$

以降では , \mathbf{p}_j を制御点 , $\mathbf{b}_n(t)$ を基底関数と呼ぶ . 制御点 \mathbf{p}_j を決定すると関節角軌道が定まる . 制御点 \mathbf{p}_j を動作計画の設計変数とする .

$j = 1, 2, \dots, N_{\text{joint}}$ 番目の関節角軌道を並べたベクトル関数は ,

$$\boldsymbol{\theta}(t) \stackrel{\text{def}}{=} \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_{N_{\text{joint}}}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{p}_1^T \mathbf{b}_n(t) \\ \mathbf{p}_2^T \mathbf{b}_n(t) \\ \vdots \\ \mathbf{p}_{N_{\text{joint}}}^T \mathbf{b}_n(t) \end{pmatrix} = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_{N_{\text{joint}}}^T \end{pmatrix} \mathbf{b}_n(t) = \mathbf{P} \mathbf{b}_n(t) \in \mathbb{R}^{N_{\text{joint}}} \quad (237)$$

ただし ,

$$\mathbf{P} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_{N_{\text{joint}}}^T \end{pmatrix} \in \mathbb{R}^{N_{\text{joint}} \times m} \quad (238)$$

⁸ 数学的帰納法で証明できる . <http://mat.fsv.cvut.cz/gcg/sbornik/prochazkova.pdf>

式 (237) は，制御点を縦に並べたベクトルとして分離して，以下のようにも表現できる．

$$\boldsymbol{\theta}(t) = \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_{N_{joint}}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{b}_n^T(t) \mathbf{p}_1 \\ \mathbf{b}_n^T(t) \mathbf{p}_2 \\ \vdots \\ \mathbf{b}_n^T(t) \mathbf{p}_{N_{joint}} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_n^T(t) & & \mathbf{O} \\ & \mathbf{b}_n^T(t) & \\ & & \ddots \\ \mathbf{O} & & & \mathbf{b}_n^T(t) \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{N_{joint}} \end{pmatrix} = \mathbf{B}_n(t) \mathbf{p} \in \mathbb{R}^{N_{joint}} \quad (239)$$

ただし，

$$\mathbf{B}_n(t) \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{b}_n^T(t) & & \mathbf{O} \\ & \mathbf{b}_n^T(t) & \\ & & \ddots \\ \mathbf{O} & & & \mathbf{b}_n^T(t) \end{pmatrix} \in \mathbb{R}^{N_{joint} \times mN_{joint}}, \quad \mathbf{p} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{N_{joint}} \end{pmatrix} \in \mathbb{R}^{mN_{joint}} \quad (240)$$

B スプラインによる関節角軌道の微分

式 (237) と式 (231) から，関節角速度軌道は次式で得られる．

$$\dot{\boldsymbol{\theta}}(t) = \mathbf{P} \dot{\mathbf{b}}_n(t) \quad (241)$$

$$= \mathbf{P} \mathbf{D} \mathbf{b}_{n-1}(t) \quad (242)$$

$$= \begin{pmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_{N_{joint}}^T \end{pmatrix} \mathbf{D} \mathbf{b}_{n-1}(t) \quad (243)$$

$$= \begin{pmatrix} \mathbf{p}_1^T \mathbf{D} \mathbf{b}_{n-1}(t) \\ \vdots \\ \mathbf{p}_{N_{joint}}^T \mathbf{D} \mathbf{b}_{n-1}(t) \end{pmatrix} \quad (244)$$

$$= \begin{pmatrix} \mathbf{b}_{n-1}^T(t) \mathbf{D}^T \mathbf{p}_1 \\ \vdots \\ \mathbf{b}_{n-1}^T(t) \mathbf{D}^T \mathbf{p}_{N_{joint}} \end{pmatrix} \quad (245)$$

$$= \begin{pmatrix} \mathbf{b}_{n-1}^T(t) \mathbf{D}^T & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{b}_{n-1}^T(t) \mathbf{D}^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{N_{joint}} \end{pmatrix} \quad (246)$$

$$= \begin{pmatrix} \mathbf{b}_{n-1}^T(t) & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{b}_{n-1}^T(t) \end{pmatrix} \begin{pmatrix} \mathbf{D}^T & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{D}^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{N_{joint}} \end{pmatrix} \quad (247)$$

$$= \mathbf{B}_{n-1}(t) \hat{\mathbf{D}}_1 \mathbf{p} \quad (248)$$

ただし，

$$\hat{\mathbf{D}}_1 = \begin{pmatrix} \mathbf{D}^T & & \mathbf{O} \\ & \mathbf{D}^T & \\ & & \ddots \\ \mathbf{O} & & & \mathbf{D}^T \end{pmatrix} \in \mathbb{R}^{mN_{joint} \times mN_{joint}} \quad (249)$$

同様にして、関節角軌道の k 階微分は次式で得られる。

$$\boldsymbol{\theta}^{(k)}(t) = \frac{d^{(k)}\boldsymbol{\theta}(t)}{dt^{(k)}} \quad (250)$$

$$= \boldsymbol{P}\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t) \quad (251)$$

$$= \boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\boldsymbol{p} \quad (252)$$

ただし、

$$\hat{\boldsymbol{D}}_k = \begin{pmatrix} (\boldsymbol{D}^k)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & (\boldsymbol{D}^k)^T \end{pmatrix} = (\hat{\boldsymbol{D}}_1)^k \in \mathbb{R}^{mN_{joint} \times mN_{joint}} \quad (253)$$

計算時間は式 (251) のほうが式 (252) より速い。

エンドエフェクタ位置姿勢拘束のタスク関数

関節角 $\boldsymbol{\theta} \in \mathbb{R}^{N_{joint}}$ からエンドエフェクタ位置姿勢 $\boldsymbol{r} \in \mathbb{R}^6$ への写像を $\boldsymbol{f}(\boldsymbol{\theta})$ で表す。

$$\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{\theta}) \quad (254)$$

関節角軌道が式 (239) で表現されるとき、エンドエフェクタ軌道は次式で表される。

$$\boldsymbol{r}(t) = \boldsymbol{f}(\boldsymbol{\theta}(t)) = \boldsymbol{f}(\boldsymbol{B}_n(t)\boldsymbol{p}) \quad (255)$$

$l = 1, 2, \dots, N_{tm}$ について、時刻 t_l でエンドエフェクタの位置姿勢が \boldsymbol{r}_l であるタスクのタスク関数は次式で表される。以降では、 t_l をタイミングと呼ぶ。

$$\boldsymbol{e}(\boldsymbol{p}, \boldsymbol{t}) \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{e}_1(\boldsymbol{p}, t) \\ \boldsymbol{e}_2(\boldsymbol{p}, t) \\ \vdots \\ \boldsymbol{e}_{N_{tm}}(\boldsymbol{p}, t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_1 - \boldsymbol{f}(\boldsymbol{\theta}(t_1)) \\ \boldsymbol{r}_2 - \boldsymbol{f}(\boldsymbol{\theta}(t_2)) \\ \vdots \\ \boldsymbol{r}_{N_{tm}} - \boldsymbol{f}(\boldsymbol{\theta}(t_{N_{tm}})) \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_1 - \boldsymbol{f}(\boldsymbol{B}_n(t_1)\boldsymbol{p}) \\ \boldsymbol{r}_2 - \boldsymbol{f}(\boldsymbol{B}_n(t_2)\boldsymbol{p}) \\ \vdots \\ \boldsymbol{r}_{N_{tm}} - \boldsymbol{f}(\boldsymbol{B}_n(t_{N_{tm}})\boldsymbol{p}) \end{pmatrix} \in \mathbb{R}^{6N_{tm}} \quad (256)$$

ただし、

$$\boldsymbol{e}_l(\boldsymbol{p}, t) \stackrel{\text{def}}{=} \boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{\theta}(t_l)) = \boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{B}_n(t_l)\boldsymbol{p}) \in \mathbb{R}^6 \quad (l = 1, 2, \dots, N_{tm}) \quad (257)$$

$$\boldsymbol{t} \stackrel{\text{def}}{=} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_{N_{tm}} \end{pmatrix} \in \mathbb{R}^{N_{tm}} \quad (258)$$

このタスクを実現する関節角軌道は、次の評価関数を最小にする制御点 \boldsymbol{p} 、タイミング \boldsymbol{t} を求めることで導出することができる。

$$F(\boldsymbol{p}, \boldsymbol{t}) \stackrel{\text{def}}{=} \frac{1}{2} \|\boldsymbol{e}(\boldsymbol{p}, \boldsymbol{t})\|^2 \quad (259)$$

$$= \frac{1}{2} \sum_{l=1}^{N_{tm}} \|\boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{\theta}(t_l))\|^2 \quad (260)$$

$$= \frac{1}{2} \sum_{l=1}^{N_{tm}} \|\boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{B}_n(t_l)\boldsymbol{p})\|^2 \quad (261)$$

また, l 番目の幾何拘束の許容誤差を $e_{tol,l} \geq 0 \in \mathbb{R}^6$ とする場合, タスク関数 $\tilde{e}_l(\mathbf{p}, t)$ は次式で表される.

$$\tilde{e}_{l,i}(\mathbf{p}, t) \stackrel{\text{def}}{=} \begin{cases} e_{l,i}(\mathbf{p}, t) - e_{tol,l,i} & e_{l,i}(\mathbf{p}, t) > e_{tol,l,i} \\ e_{l,i}(\mathbf{p}, t) + e_{tol,l,i} & e_{l,i}(\mathbf{p}, t) < -e_{tol,l,i} \\ 0 & \text{otherwise} \end{cases} \quad (i = 1, 2, \dots, 6) \quad (262)$$

$\tilde{e}_{l,i}(\mathbf{p}, t)$ は $\tilde{e}_l(\mathbf{p}, t)$ の i 番目の要素である. $e_{l,i}(\mathbf{p}, t)$ は $e(\mathbf{p}, t)$ の i 番目の要素である.

タスク関数を制御点で微分したヤコビ行列

式 (261) を目的関数とする最適化問題を Gauss-Newton 法, Levenberg-Marquardt 法や逐次二次計画法で解く場合, タスク関数 (256) のヤコビ行列が必要となる.

各時刻でのエンドエフェクタ位置姿勢拘束のタスク関数 $e_l(\mathbf{p}, t)$ の制御点 \mathbf{p} に対するヤコビ行列は次式で求められる.

$$\frac{\partial e_l(\mathbf{p}, t)}{\partial \mathbf{p}} = \frac{\partial}{\partial \mathbf{p}} \{ \mathbf{r}_l - \mathbf{f}(\mathbf{B}_n(t_l)\mathbf{p}) \} \quad (263)$$

$$= -\frac{\partial}{\partial \mathbf{p}} \mathbf{f}(\mathbf{B}_n(t_l)\mathbf{p}) \quad (264)$$

$$= -\left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}(t_l)} \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{p}} \quad (265)$$

$$= -\mathbf{J}(\boldsymbol{\theta}(t_l)) \frac{\partial}{\partial \mathbf{p}} \{ \mathbf{B}_n(t_l)\mathbf{p} \} \quad (266)$$

$$= -\mathbf{J}(\boldsymbol{\theta}(t_l)) \mathbf{B}_n(t_l) \quad (267)$$

途中の変形で, $\boldsymbol{\theta}(\mathbf{p}; t) = \mathbf{B}_n(t)\mathbf{p}$ であることを利用した. ただし,

$$\mathbf{J} \stackrel{\text{def}}{=} \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \quad (268)$$

タスク関数をタイミングで微分したヤコビ行列

各時刻でのエンドエフェクタ位置姿勢拘束のタスク関数 $e_l(\mathbf{p}, t)$ のタイミング t に対するヤコビ行列は次式で求められる.

$$\frac{\partial e_l(\mathbf{p}, t)}{\partial t_l} = \frac{\partial}{\partial t_l} \{ \mathbf{r}_l - \mathbf{f}(\mathbf{P}\mathbf{b}_n(t_l)) \} \quad (269)$$

$$= -\frac{\partial}{\partial t_l} \mathbf{f}(\mathbf{P}\mathbf{b}_n(t_l)) \quad (270)$$

$$= -\left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}(t_l)} \frac{\partial \boldsymbol{\theta}}{\partial t_l} \quad (271)$$

$$= -\mathbf{J}(\boldsymbol{\theta}(t_l)) \frac{\partial}{\partial t_l} \{ \mathbf{P}\mathbf{b}_n(t_l) \} \quad (272)$$

$$= -\mathbf{J}(\boldsymbol{\theta}(t_l)) \mathbf{P}\dot{\mathbf{b}}_n(t_l) \quad (273)$$

$$= -\mathbf{J}(\boldsymbol{\theta}(t_l)) \mathbf{P}\mathbf{D}\mathbf{b}_{n-1}(t_l) \quad (274)$$

途中の変形で, $\boldsymbol{\theta}(\mathbf{p}; t) = \mathbf{P}\mathbf{b}_n(t)$ であることを利用した.

初期・終端関節速度・加速度のタスク関数とヤコビ行列

初期，終端時刻の関節速度，加速度はゼロであるべきである．タスク関数は次式となる．

$$e_{sv}(\mathbf{p}, t) \stackrel{\text{def}}{=} \dot{\boldsymbol{\theta}}(t_s) \quad (275)$$

$$= \mathbf{B}_{n-1}(t_s) \hat{\mathbf{D}}_1 \mathbf{p} \quad (276)$$

$$= \mathbf{PD} \mathbf{b}_{n-1}(t_s) \quad (277)$$

$$e_{fv}(\mathbf{p}, t) \stackrel{\text{def}}{=} \dot{\boldsymbol{\theta}}(t_f) \quad (278)$$

$$= \mathbf{B}_{n-1}(t_f) \hat{\mathbf{D}}_1 \mathbf{p} \quad (279)$$

$$= \mathbf{PD} \mathbf{b}_{n-1}(t_f) \quad (280)$$

$$e_{sa}(\mathbf{p}, t) \stackrel{\text{def}}{=} \ddot{\boldsymbol{\theta}}(t_s) \quad (281)$$

$$= \mathbf{B}_{n-2}(t_s) \hat{\mathbf{D}}_2 \mathbf{p} \quad (282)$$

$$= \mathbf{PD}^2 \mathbf{b}_{n-2}(t_s) \quad (283)$$

$$e_{fa}(\mathbf{p}, t) \stackrel{\text{def}}{=} \ddot{\boldsymbol{\theta}}(t_f) \quad (284)$$

$$= \mathbf{B}_{n-2}(t_f) \hat{\mathbf{D}}_2 \mathbf{p} \quad (285)$$

$$= \mathbf{PD}^2 \mathbf{b}_{n-2}(t_f) \quad (286)$$

制御点で微分したヤコビ行列は次式で表される．

$$\frac{\partial e_{sv}(\mathbf{p}, t)}{\partial \mathbf{p}} = \mathbf{B}_{n-1}(t_s) \hat{\mathbf{D}}_1 \quad (287)$$

$$\frac{\partial e_{fv}(\mathbf{p}, t)}{\partial \mathbf{p}} = \mathbf{B}_{n-1}(t_f) \hat{\mathbf{D}}_1 \quad (288)$$

$$\frac{\partial e_{sa}(\mathbf{p}, t)}{\partial \mathbf{p}} = \mathbf{B}_{n-2}(t_s) \hat{\mathbf{D}}_2 \quad (289)$$

$$\frac{\partial e_{fa}(\mathbf{p}, t)}{\partial \mathbf{p}} = \mathbf{B}_{n-2}(t_f) \hat{\mathbf{D}}_2 \quad (290)$$

初期時刻，終端時刻で微分したヤコビ行列は次式で表される．

$$\frac{\partial e_{sv}(\mathbf{p}, t)}{\partial t_s} = \mathbf{PD} \frac{\partial \mathbf{b}_{n-1}(t_s)}{\partial t_s} = \mathbf{PD}^2 \mathbf{b}_{n-2}(t_s) \quad (291)$$

$$\frac{\partial e_{fv}(\mathbf{p}, t)}{\partial t_f} = \mathbf{PD} \frac{\partial \mathbf{b}_{n-1}(t_f)}{\partial t_f} = \mathbf{PD}^2 \mathbf{b}_{n-2}(t_f) \quad (292)$$

$$\frac{\partial e_{sa}(\mathbf{p}, t)}{\partial t_s} = \mathbf{PD}^2 \frac{\partial \mathbf{b}_{n-2}(t_s)}{\partial t_s} = \mathbf{PD}^3 \mathbf{b}_{n-3}(t_s) \quad (293)$$

$$\frac{\partial e_{fa}(\mathbf{p}, t)}{\partial t_f} = \mathbf{PD}^2 \frac{\partial \mathbf{b}_{n-2}(t_f)}{\partial t_f} = \mathbf{PD}^3 \mathbf{b}_{n-3}(t_f) \quad (294)$$

関節角上下限制約

式 (235) の関節角軌道定義において，

$$\mathbf{p}_j \leq \theta_{max,j} \mathbf{1}_m \quad (295)$$

のとき，B スプラインの凸包性 (式 (229)，式 (230)) より次式が成り立つ．ただし， $\mathbf{1}_m \in \mathbb{R}^m$ は全要素が 1

の m 次元ベクトルである．

$$\theta_j(t) = \sum_{i=0}^{m-1} p_{j,i} b_{i,n}(t) \quad (296)$$

$$\leq \sum_{i=0}^{m-1} \theta_{max,j} b_{i,n}(t) \quad (297)$$

$$= \theta_{max,j} \sum_{i=0}^{m-1} b_{i,n}(t) \quad (298)$$

$$= \theta_{max,j} \quad (299)$$

同様に， $\theta_{min,j} \mathbf{1}_m \leq \mathbf{p}_j$ とすれば， $\theta_{min,j} \leq \theta_j(t)$ が成り立つ．

したがって， j 番目の関節角の上下限を $\theta_{max,j}, \theta_{min,j}$ とすると，次式の制約を制御点に課すことで，関節角上下限制約を満たす関節角軌道が得られる．

$$\theta_{min,j} \mathbf{1}_m \leq \mathbf{p}_j \leq \theta_{max,j} \mathbf{1}_m \quad (j = 1, 2, \dots, N_{joint}) \quad (300)$$

つまり，

$$\hat{\mathbf{E}} \boldsymbol{\theta}_{min} \leq \mathbf{p} \leq \hat{\mathbf{E}} \boldsymbol{\theta}_{max} \quad (301)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \mathbf{p} \geq \begin{pmatrix} \hat{\mathbf{E}} \boldsymbol{\theta}_{min} \\ -\hat{\mathbf{E}} \boldsymbol{\theta}_{max} \end{pmatrix} \quad (302)$$

ただし，

$$\hat{\mathbf{E}} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{1}_m & & \mathbf{0}_m \\ & \mathbf{1}_m & \\ & & \ddots \\ \mathbf{0}_m & & \mathbf{1}_m \end{pmatrix} \in \mathbb{R}^{m N_{joint} \times N_{joint}} \quad (303)$$

これは，逐次二次計画法の中で，次式の不等式制約となる．

$$\begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \Delta \mathbf{p} \geq \begin{pmatrix} \hat{\mathbf{E}} \boldsymbol{\theta}_{min} - \mathbf{p} \\ -\hat{\mathbf{E}} \boldsymbol{\theta}_{max} + \mathbf{p} \end{pmatrix} \quad (304)$$

関節角速度・角加速度上下限制約

式 (235) と式 (231) より，関節角速度軌道，角加速度軌道は次式で表される．

$$\dot{\theta}_j(t) = \mathbf{p}_j^T \dot{\mathbf{b}}_n(t) = \mathbf{p}_j^T \mathbf{D} \mathbf{b}_{n-1}(t) = (\mathbf{D}^T \mathbf{p}_j)^T \mathbf{b}_{n-1}(t) \in \mathbb{R} \quad (t_s \leq t \leq t_f) \quad (305)$$

$$\ddot{\theta}_j(t) = \mathbf{p}_j^T \ddot{\mathbf{b}}_n(t) = \mathbf{p}_j^T \mathbf{D}^2 \mathbf{b}_{n-2}(t) = ((\mathbf{D}^2)^T \mathbf{p}_j)^T \mathbf{b}_{n-2}(t) \in \mathbb{R} \quad (t_s \leq t \leq t_f) \quad (306)$$

j 番目の関節角速度，角加速度の上限を $v_{max,j}, a_{max,j}$ とする．関節角上下限制約の導出と同様に考えると，次式の制約を制御点に課すことで，関節角速度・角加速度上下限制約を満たす関節角軌道が得られる．

$$-v_{max,j} \mathbf{1}_m \leq \mathbf{D}^T \mathbf{p}_j \leq v_{max,j} \mathbf{1}_m \quad (j = 1, 2, \dots, N_{joint}) \quad (307)$$

$$-a_{max,j} \mathbf{1}_m \leq (\mathbf{D}^2)^T \mathbf{p}_j \leq a_{max,j} \mathbf{1}_m \quad (j = 1, 2, \dots, N_{joint}) \quad (308)$$

つまり,

$$-\hat{\mathbf{E}}\mathbf{v}_{max} \leq \hat{\mathbf{D}}_1\mathbf{p} \leq \hat{\mathbf{E}}\mathbf{v}_{max} \quad (309)$$

$$\Leftrightarrow \begin{pmatrix} \hat{\mathbf{D}}_1 \\ -\hat{\mathbf{D}}_1 \end{pmatrix} \mathbf{p} \geq \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{v}_{max} \\ -\hat{\mathbf{E}}\mathbf{v}_{max} \end{pmatrix} \quad (310)$$

$$-\hat{\mathbf{E}}\mathbf{a}_{max} \leq \hat{\mathbf{D}}_2\mathbf{p} \leq \hat{\mathbf{E}}\mathbf{a}_{max} \quad (311)$$

$$\Leftrightarrow \begin{pmatrix} \hat{\mathbf{D}}_2 \\ -\hat{\mathbf{D}}_2 \end{pmatrix} \mathbf{p} \geq \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{a}_{max} \\ -\hat{\mathbf{E}}\mathbf{a}_{max} \end{pmatrix} \quad (312)$$

これは, 逐次二次計画法の中で, 次式の不等式制約となる.

$$\begin{pmatrix} \hat{\mathbf{D}}_1 \\ -\hat{\mathbf{D}}_1 \end{pmatrix} \Delta\mathbf{p} \geq \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{v}_{max} - \hat{\mathbf{D}}_1\mathbf{p} \\ -\hat{\mathbf{E}}\mathbf{v}_{max} + \hat{\mathbf{D}}_1\mathbf{p} \end{pmatrix} \quad (313)$$

$$\begin{pmatrix} \hat{\mathbf{D}}_2 \\ -\hat{\mathbf{D}}_2 \end{pmatrix} \Delta\mathbf{p} \geq \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{a}_{max} - \hat{\mathbf{D}}_2\mathbf{p} \\ -\hat{\mathbf{E}}\mathbf{a}_{max} + \hat{\mathbf{D}}_2\mathbf{p} \end{pmatrix} \quad (314)$$

タイミング上下限制約

タイミングが初期, 終端時刻の間に含まれる制約は次式で表される.

$$t_s \leq t_l \leq t_f \quad (l = 1, 2, \dots, N_{tm}) \quad (315)$$

$$\Leftrightarrow t_s \mathbf{1} \leq \mathbf{t} \leq t_f \mathbf{1} \quad (316)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \mathbf{t} \geq \begin{pmatrix} t_s \mathbf{1} \\ -t_f \mathbf{1} \end{pmatrix} \quad (317)$$

これは, 逐次二次計画法の中で, 次式の不等式制約となる.

$$\begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \Delta\mathbf{t} \geq \begin{pmatrix} t_s \mathbf{1} - \mathbf{t} \\ -t_f \mathbf{1} + \mathbf{t} \end{pmatrix} \quad (318)$$

$$(319)$$

また, タイミングの順序が入れ替わることを許容しない場合, その制約は次式で表される.

$$t_l \leq t_{l+1} \quad (l = 1, 2, \dots, N_{tm} - 1) \quad (320)$$

$$\Leftrightarrow -t_l + t_{l+1} \geq 0 \quad (l = 1, 2, \dots, N_{tm} - 1) \quad (321)$$

$$\Leftrightarrow \mathbf{D}_{tm} \mathbf{t} \geq \mathbf{0} \quad (322)$$

ただし,

$$\mathbf{D}_{tm} = \begin{pmatrix} -1 & 1 & & & \mathbf{O} \\ & -1 & 1 & & \\ & & & \ddots & \\ \mathbf{O} & & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(N_{tm}-1) \times N_{tm}} \quad (323)$$

これは, 逐次二次計画法の中で, 次式の不等式制約となる.

$$\mathbf{D}_{tm} \Delta\mathbf{t} \geq -\mathbf{D}_{tm} \mathbf{t} \quad (324)$$

関節角微分二乗積分最小化

関節角微分の二乗積分は次式で得られる．

$$F_{sqT,k}(\mathbf{p}) = \int_{t_s}^{t_f} \left\| \boldsymbol{\theta}^{(k)}(t) \right\|^2 dt \quad (325)$$

$$= \int_{t_s}^{t_f} \left\| \mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k \mathbf{p} \right\|^2 dt \quad (326)$$

$$= \int_{t_s}^{t_f} \left(\mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k \mathbf{p} \right)^T \left(\mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k \mathbf{p} \right) dt \quad (327)$$

$$= \mathbf{p}^T \left\{ \int_{t_s}^{t_f} \left(\mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k \right)^T \mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k dt \right\} \mathbf{p} \quad (328)$$

$$= \mathbf{p}^T \mathbf{H}_k \mathbf{p} \quad (329)$$

ただし，

$$\mathbf{H}_k = \int_{t_s}^{t_f} \left(\mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k \right)^T \mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k dt \quad (330)$$

$$\mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k = \begin{pmatrix} \mathbf{b}_{n-k}^T(t) & \mathbf{O} \\ & \ddots \\ \mathbf{O} & \mathbf{b}_{n-k}^T(t) \end{pmatrix} \begin{pmatrix} (\mathbf{D}^k)^T & \mathbf{O} \\ & \ddots \\ \mathbf{O} & (\mathbf{D}^k)^T \end{pmatrix} \quad (331)$$

$$= \begin{pmatrix} \mathbf{b}_{n-k}^T(t) (\mathbf{D}^k)^T & \mathbf{O} \\ & \ddots \\ \mathbf{O} & \mathbf{b}_{n-k}^T(t) (\mathbf{D}^k)^T \end{pmatrix} \quad (332)$$

$$= \begin{pmatrix} \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T & \mathbf{O} \\ & \ddots \\ \mathbf{O} & \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T \end{pmatrix} \quad (333)$$

$$\left(\mathbf{B}_{n-k}(t) \hat{\mathbf{D}}_k \right)^T \mathbf{B}_{n-k}(t) = \begin{pmatrix} \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T & \mathbf{O} \\ & \ddots \\ \mathbf{O} & \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T \end{pmatrix}^T \begin{pmatrix} \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T & \mathbf{O} \\ & \ddots \\ \mathbf{O} & \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T \end{pmatrix} \quad (334)$$

$$= \begin{pmatrix} \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right) \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T & & \\ & \ddots & \\ \mathbf{O} & & \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right) \left(\mathbf{D}^k \mathbf{b}_{n-k}(t) \right)^T \end{pmatrix} \quad (335)$$

これを逐次二次計画問題において，二次形式の正則化項として目的関数に加えることで，滑らかな動作が生成されることが期待される．

動作期間の最小化

動作期間 $(t_f - t_s)$ の二乗は次式で表される．

$$F_{duration}(\mathbf{t}) = |t_1 - t_{N_{tm}}|^2 \quad (336)$$

$$= \mathbf{t}^T \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \mathbf{t} \quad (337)$$

ただし, 初期時刻 $t_s = t_1$, 終端時刻 $t_f = t_{N_{tm}}$ がタイミングベクトル t の最初, 最後の要素であるとする. これを逐次二次計画問題において, 二次形式の正則化項として目的関数に加えることで, 短時間でタスクを実現する動作が生成されることが期待される.

4.2.2 B スプラインを用いた関節軌道生成の実装

bspline-configuration-task

[class]

```

:super    propertied-object
:slots    (_robot robot instance)
           (_control-vector  $\mathbf{p}$ )
           (_timing-vector  $\mathbf{t}$ )
           (_num-kin  $N_{kin} := |\mathcal{T}^{kin-trg}| = |\mathcal{T}^{kin-att}|$ )
           (_num-joint  $N_{joint} := |\mathcal{J}|$ )
           (_num-control-point  $N_{ctrl}$ )
           (_num-timing  $N_{tm}$ )
           (_bspline-order B-spline order,  $n$ )
           (_dim-control-vector  $\dim(\mathbf{p})$ )
           (_dim-timing-vector  $\dim(\mathbf{t})$ )
           (_dim-config  $\dim(\mathbf{q})$ )
           (_dim-task  $\dim(\mathbf{e})$ )
           (_num-collision  $N_{col} :=$  number of collision check pairs)
           (_stationery-start-finish-task-scale  $k_{stat}$ )
           (_first-diff-square-integration-regular-scale  $k_{sqr,1}$ )
           (_second-diff-square-integration-regular-scale  $k_{sqr,2}$ )
           (_third-diff-square-integration-regular-scale  $k_{sqr,3}$ )
           (_motion-duration-regular-scale  $k_{duration}$ )
           (_norm-regular-scale-max  $k_{max,p}$ )
           (_norm-regular-scale-offset  $k_{off,p}$ )
           (_timing-norm-regular-scale-max  $k_{max,t}$ )
           (_timing-norm-regular-scale-offset  $k_{off,t}$ )
           (_joint-list  $\mathcal{J}$ )
           (_start-time  $t_s$ )
           (_finish-time  $t_f$ )
           (_kin-time-list  $\{t_1^{kin-tm}, t_2^{kin-tm}, \dots, t_{N_{kin}}^{kin-tm}\}$ )
           (_kin-variable-timing-list list of bool. t for variable timing.)
           (_kin-target-coords-list  $\mathcal{T}^{kin-trg}$ )
           (_kin-attention-coords-list  $\mathcal{T}^{kin-att}$ )
           (_kin-pos-tolerance-list list of position tolerance  $e_{tol,pos}$  [m])
           (_kin-rot-tolerance-list list of rotation tolerance  $e_{tol,rot}$  [rad])
           (_joint-angle-margin margin of  $\theta$  [deg] [mm])
           (_collision-pair-list list of bodyset-link or body pair)
           (_keep-timing-order? whether to keep order of timing  $\mathbf{t}$  or not)
           (_bspline-matrix buffer for  $\mathbf{B}_n(t)$ )

```

(_diff-mat buffer for D^k)
 (_diff-mat-list buffer for $\{D^1, D^2, \dots, D^K\}$)
 (_extended-diff-mat-list buffer for $\{\hat{D}_1, \hat{D}_2, \dots, \hat{D}_K\}$)
 (_task-jacobi buffer for $\frac{\partial e}{\partial q}$)
 (_regular-mat buffer for W_{reg})
 (_regular-vec buffer for v_{reg})

B スプラインを利用した軌道生成のためのコンフィギュレーション q とタスク関数 $e(q)$ のクラス .

コンフィギュレーション q の取得・更新, タスク関数 $e(q)$ の取得, タスク関数のヤコビ行列 $\frac{\partial e(q)}{\partial q}$ の取得, コンフィギュレーションの等式・不等式制約 A, b, C, d の取得のためのメソッドが定義されている .

コンフィギュレーション・タスク関数を定めるために, 初期化時に以下を与える

- ロボット

robot ロボットのインスタンス

joint-list \mathcal{J} 関節

- B スプラインのパラメータ

start-time t_s B スプラインの使用区間の初期時刻

finish-time t_f B スプラインの使用区間の終端時刻

num-control-point N_{ctrl} 制御点の個数

bspline-order n B スプラインの次数

- 幾何拘束

kin-target-coords-list $\mathcal{T}^{kin-try}$ 幾何到達目標位置姿勢リスト

kin-attention-coords-list $\mathcal{T}^{kin-att}$ 幾何到達着目位置姿勢リスト

kin-time-list $\{t_1^{kin-tm}, t_2^{kin-tm}, \dots, t_{N_{kin}}^{kin-tm}\}$ 幾何到達タイミングリスト

kin-variable-timing-list 幾何到達タイミングが可変か (t), 固定か (nil) のリスト . このリスト内の t の個数がタイミング t の次元 N_{tm} となる .

コンフィギュレーション q は以下から構成される .

$$q := \begin{pmatrix} p \\ t \end{pmatrix} \quad (338)$$

$p \in \mathbb{R}^{N_{ctrl}N_{joint}}$ 制御点 (B スプライン基底関数の山の高さ) [rad] [m]

$t \in \mathbb{R}^{N_{tm}}$ タイミング (幾何拘束タスクの課される時刻) [sec]

タスク関数 $e(q)$ は以下から構成される .

$$e(q) := \begin{pmatrix} e^{kin}(q) \\ e^{stat}(q) \end{pmatrix} \in \mathbb{R}^{6N_{kin}+4N_{joint}} \quad (339)$$

$e^{kin}(q) \in \mathbb{R}^{6N_{kin}}$ 幾何到達拘束 [rad] [m]

$e^{stat}(q) \in \mathbb{R}^{4N_{joint}}$ 初期, 終端時刻静止拘束 [rad][rad/s][rad/s²][m][m/s][m/s²]

| | |
|---|----------|
| :init \mathcal{E}_{key} (name) | [method] |
| (robot) (joint-list (send robot :joint-list)) (start-time 0.0) (finish-time 10.0) (num-control-point 10) (bspline-order 3) (kin-time-list) (kin-variable-timing-list (make-list (length kin-time-list) :initial-element nil)) (kin-target-coords-list) (kin-attention-coords-list) (kin-pos-tolerance-list (make-list (length kin-time-list) :initial-element 0.0)) (kin-rot-tolerance-list (make-list (length kin-time-list) :initial-element 0.0)) (joint-angle-margin 3.0) (collision-pair-list) (keep-timing-order? t) (stationery-start-finish-task-scale 0.0) (first-diff-square-integration-regular-scale 0.0) (second-diff-square-integration-regular-scale 0.0) (third-diff-square-integration-regular-scale 0.0) (motion-duration-regular-scale 0.0) (norm-regular-scale-max 1.000000e-05) (norm-regular-scale-offset 1.000000e-07) (timing-norm-regular-scale-max 1.000000e-05) (timing-norm-regular-scale-offset 1.000000e-07) | |
| Initialize instance | |
| :robot | [method] |
| return robot instance | |
| :joint-list | [method] |
| return \mathcal{J} | |
| :num-kin | [method] |
| return $N_{kin} := \mathcal{T}^{kin-trg} = \mathcal{T}^{kin-att} $ | |
| :num-joint | [method] |
| return $N_{joint} := \mathcal{J} $ | |
| :num-control-point | [method] |
| return N_{ctrl} | |
| :num-timing | [method] |
| return N_{tm} | |
| :num-collision | [method] |
| return $N_{col} :=$ number of collision check pairs | |
| :dim-config | [method] |
| return $dim(\mathbf{q}) := dim(\mathbf{p}) + dim(\mathbf{t}) = N_{ctrl}N_{joint} + N_{tm}$ | |

:dim-task [method]

return $\dim(\mathbf{e}) := \dim(\mathbf{e}^{kin}) + \dim(\mathbf{e}^{stat}) = 6N_{kin} + 4N_{joint}$

:control-vector [method]

return control vector \mathbf{p}

:timing-vector [method]

return timing vector \mathbf{t}

:config-vector [method]

return $\mathbf{q} := \begin{pmatrix} \mathbf{p} \\ \mathbf{t} \end{pmatrix}$

:set-control-vector *control-vector-new* *ℰkey* (*relative?* *nil*) [method]

Set \mathbf{p} .

:set-timing-vector *timing-vector-new* *ℰkey* (*relative?* *nil*) [method]

Set \mathbf{t} .

:set-config *config-new* *ℰkey* (*relative?* *nil*) [method]

Set \mathbf{q} .

:bspline-vector *tm* *ℰkey* (*order-offset* 0) [method]

$$\mathbf{b}_n(t) := \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{N_{ctrl}-1,n}(t) \end{pmatrix} \in \mathbb{R}^{N_{ctrl}} \quad (340)$$

return $\mathbf{b}_n(t)$

:bspline-matrix *tm* *ℰkey* (*order-offset* 0) [method]

$$\mathbf{B}_n(t) := \begin{pmatrix} \mathbf{b}_n^T(t) & & & \mathbf{O} \\ & \mathbf{b}_n^T(t) & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{b}_n^T(t) \end{pmatrix} \in \mathbb{R}^{N_{joint} \times N_{ctrl} N_{joint}} \quad (341)$$

return $\mathbf{B}_n(t)$

:differential-matrix *ℰkey* (*diff-order* 1) [method]

$$\mathbf{D} := \frac{1}{h} \begin{pmatrix} 1 & -1 & & & \mathbf{O} \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -1 \\ \mathbf{O} & & & & 1 \end{pmatrix} \in \mathbb{R}^{N_{ctrl} \times N_{ctrl}} \quad (342)$$

return \mathbf{D}^k

:extended-differential-matrix *ℰkey (diff-order 1)*

[method]

$$\hat{\mathbf{D}}_k := \begin{pmatrix} (\mathbf{D}^k)^T & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & (\mathbf{D}^k)^T \end{pmatrix} \in \mathbb{R}^{N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}} \quad (343)$$

return $\hat{\mathbf{D}}_k$

:bspline-differential-matrix *tm ℰkey (diff-order 1)*

[method]

return $\mathbf{B}_{n-k}(t)\hat{\mathbf{D}}_k \in \mathbb{R}^{N_{joint} \times N_{ctrl}N_{joint}}$

:control-matrix

[method]

$$\mathbf{P} := \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_{n_{joint}}^T \end{pmatrix} \in \mathbb{R}^{N_{joint} \times N_{ctrl}} \quad (344)$$

return \mathbf{P}

:theta *tm*

[method]

return $\boldsymbol{\theta}(t) = \mathbf{B}_n(t)\mathbf{p}$ [rad][m]

:theta-dot *tm ℰkey (diff-order 1)*

[method]

return $\boldsymbol{\theta}^{(k)}(t) = \frac{d^{(k)}\boldsymbol{\theta}(t)}{dt^{(k)}} = \mathbf{P}\mathbf{D}^k\mathbf{b}_{n-k}(t)$ [rad/s^k][m/s^k]

:theta-dot-numerical *tm ℰkey (diff-order 1)*

[method]

(delta-time 0.05)

return $\boldsymbol{\theta}^{(k)}(t) = \frac{d^{(k)}\boldsymbol{\theta}(t)}{dt^{(k)}} = \frac{\boldsymbol{\theta}^{(k-1)}(t + \Delta t) - \boldsymbol{\theta}^{(k-1)}(t)}{\Delta t}$ [rad/s^k][m/s^k]

:apply-theta-to-robot *tm*

[method]

apply $\boldsymbol{\theta}(t)$ to robot.

:kin-target-coords-list

[method]

$$\mathbf{T}_m^{kin-trg} = \{\mathbf{p}_l^{kin-trg}, \mathbf{R}_l^{kin-trg}\} \quad (l = 1, 2, \dots, N_{kin}) \quad (345)$$

return $\mathcal{T}^{kin-trg} := \{T_1^{kin-trg}, T_2^{kin-trg}, \dots, T_{N_{kin}}^{kin-trg}\}$

:kin-attention-coords-list

[method]

$$\mathbf{T}_m^{kin-att} = \{\mathbf{p}_l^{kin-att}, \mathbf{R}_l^{kin-att}\} \quad (l = 1, 2, \dots, N_{kin}) \quad (346)$$

return $\mathcal{T}^{kin-att} := \{T_1^{kin-att}, T_2^{kin-att}, \dots, T_{N_{kin}}^{kin-att}\}$

:kin-start-time

[method]

return $t_s^{kin} := t_1^{kin-tm}$

:kin-finish-time [method]

return $t_f^{kin} := t_{N_{kin}}^{kin-tm}$

:motion-duration [method]

return $(t_{N_{kin}}^{kin-tm} - t_1^{kin-tm})$

:kinematics-task-value $\mathcal{E}key$ (update? t) [method]

$$\mathbf{e}^{kin}(\mathbf{q}) = \mathbf{e}^{kin}(\mathbf{p}, t) \quad (347)$$

$$= \begin{pmatrix} \mathbf{e}_1^{kin}(\mathbf{p}, t) \\ \mathbf{e}_2^{kin}(\mathbf{p}, t) \\ \vdots \\ \mathbf{e}_{N_{kin}}^{kin}(\mathbf{p}, t) \end{pmatrix} \quad (348)$$

$$\mathbf{e}_l^{kin}(\mathbf{p}, t) = \begin{pmatrix} \mathbf{p}_l^{kin-trg} - \mathbf{p}_l^{kin-att}(\mathbf{p}, t) \\ a(\mathbf{R}_l^{kin-trg} \mathbf{R}_l^{kin-att}(\mathbf{p}, t)^T) \end{pmatrix} \in \mathbb{R}^6 \quad (l = 1, 2, \dots, N_{kin}) \quad (349)$$

$a(\mathbf{R})$ は姿勢行列 \mathbf{R} の等価角軸ベクトルを表す .

return $\mathbf{e}^{kin}(\mathbf{q}) \in \mathbb{R}^{6N_{kin}}$

:stationery-start-finish-task-value $\mathcal{E}key$ (update? t) [method]

$$\mathbf{e}^{stat}(\mathbf{q}) = \mathbf{e}^{stat}(\mathbf{p}, t) \quad (350)$$

$$= \begin{pmatrix} \mathbf{e}_{sv}^{stat}(\mathbf{p}, t) \\ \mathbf{e}_{fv}^{stat}(\mathbf{p}, t) \\ \mathbf{e}_{sa}^{stat}(\mathbf{p}, t) \\ \mathbf{e}_{fa}^{stat}(\mathbf{p}, t) \end{pmatrix} \quad (351)$$

$$\mathbf{e}_{sv}^{stat}(\mathbf{p}, t) := \dot{\boldsymbol{\theta}}(t_s^{kin}) \quad (352)$$

$$\mathbf{e}_{fv}^{stat}(\mathbf{p}, t) := \dot{\boldsymbol{\theta}}(t_f^{kin}) \quad (353)$$

$$\mathbf{e}_{sa}^{stat}(\mathbf{p}, t) := \ddot{\boldsymbol{\theta}}(t_s^{kin}) \quad (354)$$

$$\mathbf{e}_{fa}^{stat}(\mathbf{p}, t) := \ddot{\boldsymbol{\theta}}(t_f^{kin}) \quad (355)$$

return $\mathbf{e}^{stat}(\mathbf{q}) \in \mathbb{R}^{4N_{joint}}$

:task-value $\mathcal{E}key$ (update? t) [method]

return $\mathbf{e}(\mathbf{q}) := \begin{pmatrix} \mathbf{e}^{kin}(\mathbf{q}) \\ k_{stat} \mathbf{e}^{stat}(\mathbf{q}) \end{pmatrix} \in \mathbb{R}^{6N_{kin} + 4N_{joint}}$

:kinematics-task-jacobian-with-control-vector [method]

式 (267) より , タスク関数 \mathbf{e}^{kin} を制御点 \mathbf{p} で微分したヤコビ行列は次式で得られる .

$$\frac{\partial \mathbf{e}^{kin}}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial \mathbf{e}_1^{kin}}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{e}_2^{kin}}{\partial \mathbf{p}} \\ \vdots \\ \frac{\partial \mathbf{e}_{N_{kin}}^{kin}}{\partial \mathbf{p}} \end{pmatrix} \quad (356)$$

$$\frac{\partial \mathbf{e}_l^{kin}}{\partial \mathbf{p}} = -\mathbf{J}^{kin-att}(\boldsymbol{\theta}(t_l^{kin-tm})) \mathbf{B}_n(t_l^{kin-tm}) \quad (l = 1, 2, \dots, N_{kin}) \quad (357)$$

$$\text{return } \frac{\partial \mathbf{e}^{kin}}{\partial \mathbf{p}} \in \mathbb{R}^{6N_{kin} \times N_{ctrl}N_{joint}}$$

:kinematics-task-jacobian-with-timing-vector

[method]

式 (274) より , タスク関数 e^{kin} をタイミング t で微分したヤコビ行列は次式で得られる .

$$\frac{\partial \mathbf{e}^{kin}}{\partial \mathbf{t}} = \begin{pmatrix} \frac{\partial \mathbf{e}_1^{kin}}{\partial \mathbf{t}} \\ \frac{\partial \mathbf{e}_2^{kin}}{\partial \mathbf{t}} \\ \vdots \\ \frac{\partial \mathbf{e}_{N_{kin}}^{kin}}{\partial \mathbf{t}} \end{pmatrix} \quad (358)$$

$\frac{\partial \mathbf{e}_i^{kin}}{\partial \mathbf{t}}$ の i 番目の列ベクトル $\left[\frac{\partial \mathbf{e}_i^{kin}}{\partial \mathbf{t}} \right]_i \in \mathbb{R}^6$ は次式で表される ($i = 1, 2, \dots, N_{tm}$) .

$$\left[\frac{\partial \mathbf{e}_i^{kin}}{\partial \mathbf{t}} \right]_i = \begin{cases} -\mathbf{J}^{kin-att}(\boldsymbol{\theta}(t_l^{kin-tm})) \mathbf{P} \mathbf{D} \mathbf{b}_{n-1}(t_l^{kin-tm}) & t_l^{kin-tm} \text{ and } t_i \text{ is identical} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (359)$$

$$\text{return } \frac{\partial \mathbf{e}^{kin}}{\partial \mathbf{t}} \in \mathbb{R}^{6N_{kin} \times N_{tm}}$$

:stationery-start-finish-task-jacobian-with-control-vector

[method]

式 (287) , 式 (288) , 式 (289) , 式 (290) より , タスク関数 e^{stat} を制御点 \mathbf{p} で微分したヤコビ行列は次式で得られる .

$$\frac{\partial \mathbf{e}^{stat}}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial \mathbf{e}_{sv}^{stat}}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{e}_{fv}^{stat}}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{e}_{sa}^{stat}}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{e}_{fa}^{stat}}{\partial \mathbf{p}} \end{pmatrix} \quad (360)$$

$$\frac{\partial \mathbf{e}_{sv}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{p}} = \mathbf{B}_{n-1}(t_s^{kin}) \hat{\mathbf{D}}_1 \quad (361)$$

$$\frac{\partial \mathbf{e}_{fv}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{p}} = \mathbf{B}_{n-1}(t_f^{kin}) \hat{\mathbf{D}}_1 \quad (362)$$

$$\frac{\partial \mathbf{e}_{sa}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{p}} = \mathbf{B}_{n-2}(t_s^{kin}) \hat{\mathbf{D}}_2 \quad (363)$$

$$\frac{\partial \mathbf{e}_{fa}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{p}} = \mathbf{B}_{n-2}(t_f^{kin}) \hat{\mathbf{D}}_2 \quad (364)$$

$$\text{return } \frac{\partial \mathbf{e}^{stat}}{\partial \mathbf{p}} \in \mathbb{R}^{4N_{joint} \times N_{ctrl}N_{joint}}$$

:stationery-start-finish-task-jacobian-with-timing-vector

[method]

式 (291) , 式 (292) , 式 (293) , 式 (294) より , タスク関数 e^{stat} をタイミング t で微分したヤコビ行列は次式で得られる .

$$\frac{\partial \mathbf{e}^{stat}}{\partial \mathbf{t}} = \begin{pmatrix} \frac{\partial \mathbf{e}_{sv}^{stat}}{\partial \mathbf{t}} \\ \frac{\partial \mathbf{e}_{fv}^{stat}}{\partial \mathbf{t}} \\ \frac{\partial \mathbf{e}_{sa}^{stat}}{\partial \mathbf{t}} \\ \frac{\partial \mathbf{e}_{fa}^{stat}}{\partial \mathbf{t}} \end{pmatrix} \quad (365)$$

$\frac{\partial \mathbf{e}_x^{stat}}{\partial \mathbf{t}}$ の i 番目の列ベクトル $\left[\frac{\partial \mathbf{e}_x^{stat}}{\partial \mathbf{t}} \right]_i \in \mathbb{R}^{N_{joint}}$ は次式で表される ($x \in \{sv, fv, sa, fa\}, i = 1, 2, \dots, N_{tm}$) .

$$\left[\frac{\partial \mathbf{e}_{sv}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{t}} \right]_i = \begin{cases} PD^2 \mathbf{b}_{n-2}(t_s^{kin}) & t_s^{kin} \text{ and } t_i \text{ is identical} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (366)$$

$$\left[\frac{\partial \mathbf{e}_{fv}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{t}} \right]_i = \begin{cases} PD^2 \mathbf{b}_{n-2}(t_f^{kin}) & t_f^{kin} \text{ and } t_i \text{ is identical} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (367)$$

$$\left[\frac{\partial \mathbf{e}_{sa}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{t}} \right]_i = \begin{cases} PD^3 \mathbf{b}_{n-3}(t_s^{kin}) & t_s^{kin} \text{ and } t_i \text{ is identical} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (368)$$

$$\left[\frac{\partial \mathbf{e}_{fa}^{stat}(\mathbf{p}, \mathbf{t})}{\partial \mathbf{t}} \right]_i = \begin{cases} PD^3 \mathbf{b}_{n-3}(t_f^{kin}) & t_f^{kin} \text{ and } t_i \text{ is identical} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (369)$$

return $\frac{\partial \mathbf{e}^{stat}}{\partial \mathbf{t}} \in \mathbb{R}^{4N_{joint} \times N_{tm}}$

:task-jacobian

[method]

$$\frac{\partial \mathbf{e}}{\partial \mathbf{q}} = \begin{matrix} 6N_{kin} & N_{ctrl}N_{joint} & N_{tm} \\ 4N_{joint} & \begin{pmatrix} \frac{\partial \mathbf{e}^{kin}}{\partial \mathbf{p}} & \frac{\partial \mathbf{e}^{kin}}{\partial \mathbf{t}} \\ k_{stat} \frac{\partial \mathbf{e}^{stat}}{\partial \mathbf{p}} & k_{stat} \frac{\partial \mathbf{e}^{stat}}{\partial \mathbf{t}} \end{pmatrix} \end{matrix} \quad (370)$$

return $\frac{\partial \mathbf{e}}{\partial \mathbf{q}} = \mathbb{R}^{(6N_{kin} + 4N_{joint}) \times (N_{ctrl}N_{joint} + N_{tm})}$

:theta-max-vector \mathcal{E}_{key} (update? nil)

[method]

return $\boldsymbol{\theta}_{max} \in \mathbb{R}^{N_{joint}}$

:theta-min-vector \mathcal{E}_{key} (update? nil)

[method]

return $\boldsymbol{\theta}_{min} \in \mathbb{R}^{N_{joint}}$

:theta-inequality-constraint-matrix \mathcal{E}_{key} (update? nil)

[method]

式 (301) より , 関節角度上下制限約は次式で表される .

$$\hat{\mathbf{E}}\boldsymbol{\theta}_{min} \leq \mathbf{p} + \Delta\mathbf{p} \leq \hat{\mathbf{E}}\boldsymbol{\theta}_{max} \quad (371)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \Delta\mathbf{p} \geq \begin{pmatrix} \hat{\mathbf{E}}\boldsymbol{\theta}_{min} - \mathbf{p} \\ -\hat{\mathbf{E}}\boldsymbol{\theta}_{max} + \mathbf{p} \end{pmatrix} \quad (372)$$

$$\Leftrightarrow \mathbf{C}_\theta \Delta\mathbf{p} \geq \mathbf{d}_\theta \quad (373)$$

ただし ,

$$\hat{\mathbf{E}} := \begin{pmatrix} \mathbf{1}_{N_{ctrl}} & & \mathbf{0}_{N_{ctrl}} \\ & \mathbf{1}_{N_{ctrl}} & \\ & & \ddots \\ \mathbf{0}_{N_{ctrl}} & & \mathbf{1}_{N_{ctrl}} \end{pmatrix} \in \mathbb{R}^{N_{ctrl}N_{joint} \times N_{joint}} \quad (374)$$

$\mathbf{1}_{N_{ctrl}} \in \mathbb{R}^{N_{ctrl}}$ は全要素が 1 の N_{ctrl} 次元ベクトルである .

return $\mathbf{C}_\theta := \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}}$

:theta-inequality-constraint-vector \mathcal{E}_{key} (update? t)

[method]

return $\mathbf{d}_\theta := \begin{pmatrix} \hat{\mathbf{E}}\boldsymbol{\theta}_{min} - \mathbf{p} \\ -\hat{\mathbf{E}}\boldsymbol{\theta}_{max} + \mathbf{p} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint}}$

:velocity-max-vector $\mathcal{E}key$ (*update?* *nil*) [method]
 return $\mathbf{v}_{max} \in \mathbb{R}^{N_{joint}}$

:velocity-inequality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]
 式 (309) より, 関節速度上下限制約は次式で表される .

$$-\hat{\mathbf{E}}\mathbf{v}_{max} \leq \hat{\mathbf{D}}_1(\mathbf{p} + \Delta\mathbf{p}) \leq \hat{\mathbf{E}}\mathbf{v}_{max} \quad (375)$$

$$\Leftrightarrow \begin{pmatrix} \hat{\mathbf{D}}_1 \\ -\hat{\mathbf{D}}_1 \end{pmatrix} \Delta\mathbf{p} \geq \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{v}_{max} - \hat{\mathbf{D}}_1\mathbf{p} \\ -\hat{\mathbf{E}}\mathbf{v}_{max} + \hat{\mathbf{D}}_1\mathbf{p} \end{pmatrix} \quad (376)$$

$$\Leftrightarrow \mathbf{C}_{\dot{\theta}}\Delta\mathbf{p} \geq \mathbf{d}_{\dot{\theta}} \quad (377)$$

$$\text{return } \mathbf{C}_{\dot{\theta}} := \begin{pmatrix} \hat{\mathbf{D}}_1 \\ -\hat{\mathbf{D}}_1 \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}}$$

:velocity-inequality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\text{return } \mathbf{d}_{\dot{\theta}} := \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{v}_{max} - \hat{\mathbf{D}}_1\mathbf{p} \\ -\hat{\mathbf{E}}\mathbf{v}_{max} + \hat{\mathbf{D}}_1\mathbf{p} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint}}$$

:acceleration-max-vector $\mathcal{E}key$ (*update?* *nil*) [method]

$$\text{return } \mathbf{a}_{max} \in \mathbb{R}^{N_{joint}}$$

:acceleration-inequality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

式 (311) より, 関節加速度上下限制約は次式で表される .

$$-\hat{\mathbf{E}}\mathbf{a}_{max} \leq \hat{\mathbf{D}}_2(\mathbf{p} + \Delta\mathbf{p}) \leq \hat{\mathbf{E}}\mathbf{a}_{max} \quad (378)$$

$$\Leftrightarrow \begin{pmatrix} \hat{\mathbf{D}}_2 \\ -\hat{\mathbf{D}}_2 \end{pmatrix} \Delta\mathbf{p} \geq \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{a}_{max} - \hat{\mathbf{D}}_2\mathbf{p} \\ -\hat{\mathbf{E}}\mathbf{a}_{max} + \hat{\mathbf{D}}_2\mathbf{p} \end{pmatrix} \quad (379)$$

$$\Leftrightarrow \mathbf{C}_{\ddot{\theta}}\Delta\mathbf{p} \geq \mathbf{d}_{\ddot{\theta}} \quad (380)$$

$$\text{return } \mathbf{C}_{\ddot{\theta}} := \begin{pmatrix} \hat{\mathbf{D}}_2 \\ -\hat{\mathbf{D}}_2 \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}}$$

:acceleration-inequality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\text{return } \mathbf{d}_{\ddot{\theta}} := \begin{pmatrix} -\hat{\mathbf{E}}\mathbf{a}_{max} - \hat{\mathbf{D}}_2\mathbf{p} \\ -\hat{\mathbf{E}}\mathbf{a}_{max} + \hat{\mathbf{D}}_2\mathbf{p} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint}}$$

:control-vector-inequality-constraint-matrix $\mathcal{E}key$ (*update?* *nil*) [method]

$$\begin{cases} \mathbf{C}_{\theta}\Delta\mathbf{p} \geq \mathbf{d}_{\theta} \\ \mathbf{C}_{\dot{\theta}}\Delta\mathbf{p} \geq \mathbf{d}_{\dot{\theta}} \\ \mathbf{C}_{\ddot{\theta}}\Delta\mathbf{p} \geq \mathbf{d}_{\ddot{\theta}} \end{cases} \quad (381)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{C}_{\theta} \\ \mathbf{C}_{\dot{\theta}} \\ \mathbf{C}_{\ddot{\theta}} \end{pmatrix} \Delta\mathbf{p} \geq \begin{pmatrix} \mathbf{d}_{\theta} \\ \mathbf{d}_{\dot{\theta}} \\ \mathbf{d}_{\ddot{\theta}} \end{pmatrix} \quad (382)$$

$$\Leftrightarrow \mathbf{C}_p\Delta\mathbf{p} \geq \mathbf{d}_p \quad (383)$$

$$\text{return } \mathbf{C}_p := \begin{pmatrix} \mathbf{C}_{\theta} \\ \mathbf{C}_{\dot{\theta}} \\ \mathbf{C}_{\ddot{\theta}} \end{pmatrix} \in \mathbb{R}^{N_{p-ineq} \times \dim(\mathbf{p})}$$

:control-vector-inequality-constraint-vector $\mathcal{E}key$ (*update?* *t*) [method]

$$\text{return } \mathbf{d}_p := \begin{pmatrix} \mathbf{d}_\theta \\ \mathbf{d}_{\dot{\theta}} \\ \mathbf{d}_{\ddot{\theta}} \end{pmatrix} \in \mathbb{R}^{N_{p-ineq}}$$

:timing-vector-inequality-constraint-matrix \mathcal{E}_{key} (*update?* *nil*) [method]

式 (316) より, タイミングが B スプラインの初期, 終端時刻の間に含まれる制約は次式で表される.

$$t_s \mathbf{1} \leq \mathbf{t} + \Delta \mathbf{t} \leq t_f \mathbf{1} \quad (384)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \Delta \mathbf{t} \geq \begin{pmatrix} t_s \mathbf{1} - \mathbf{t} \\ -t_f \mathbf{1} + \mathbf{t} \end{pmatrix} \quad (385)$$

また, 式 (322) より, タイミングの順序が入れ替わることを許容しない場合, その制約は次式で表される.

$$\mathbf{D}_{tm}(\mathbf{t} + \Delta \mathbf{t}) \geq \mathbf{0} \quad (386)$$

$$\Leftrightarrow \mathbf{D}_{tm} \Delta \mathbf{t} \geq -\mathbf{D}_{tm} \mathbf{t} \quad (387)$$

ただし,

$$\mathbf{D}_{tm} = \begin{pmatrix} -1 & 1 & & & \mathbf{0} \\ & -1 & 1 & & \\ & & & \ddots & \\ \mathbf{0} & & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(N_{tm}-1) \times N_{tm}} \quad (388)$$

これらを合わせると,

$$\begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{D}_{tm} \end{pmatrix} \Delta \mathbf{t} \geq \begin{pmatrix} t_s \mathbf{1} - \mathbf{t} \\ -t_f \mathbf{1} + \mathbf{t} \\ -\mathbf{D}_{tm} \mathbf{t} \end{pmatrix} \Leftrightarrow \mathbf{C}_t \Delta \mathbf{p} \geq \mathbf{d}_t \quad (389)$$

$$\text{return } \mathbf{C}_t := \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{D}_{tm} \end{pmatrix} \in \mathbb{R}^{(3N_{tm}-1) \times \dim(\mathbf{t})}$$

:timing-vector-inequality-constraint-vector \mathcal{E}_{key} (*update?* *t*) [method]

$$\text{return } \mathbf{d}_t := \begin{pmatrix} t_s \mathbf{1} - \mathbf{t} \\ -t_f \mathbf{1} + \mathbf{t} \\ -\mathbf{D}_{tm} \mathbf{t} \end{pmatrix} \in \mathbb{R}^{(3N_{tm}-1)}$$

:config-inequality-constraint-matrix \mathcal{E}_{key} (*update?* *nil*) [method]
(*update-collision?* *nil*)

$$\begin{cases} \mathbf{C}_p \Delta \mathbf{p} \geq \mathbf{d}_p \\ \mathbf{C}_t \Delta \mathbf{t} \geq \mathbf{d}_t \end{cases} \quad (390)$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{C}_p & \\ & \mathbf{C}_t \end{pmatrix} \begin{pmatrix} \Delta \mathbf{p} \\ \Delta \mathbf{t} \end{pmatrix} \geq \begin{pmatrix} \mathbf{d}_p \\ \mathbf{d}_t \end{pmatrix} \quad (391)$$

$$\Leftrightarrow \mathbf{C} \Delta \mathbf{q} \geq \mathbf{d} \quad (392)$$

$$\text{return } \mathbf{C} := \begin{pmatrix} \mathbf{C}_p & \\ & \mathbf{C}_t \end{pmatrix} \in \mathbb{R}^{N_{ineq} \times \dim(\mathbf{q})}$$

[illegible]
$$\text{return } \mathbf{d} := \begin{pmatrix} \mathbf{d}_p \\ \mathbf{d}_t \end{pmatrix} \in \mathbb{R}^{N_{ineq}}$$

```

:config-equality-constraint-matrix ℰkey (update? nil) [method]
  return  $\mathbf{A} \in \mathbb{R}^{0 \times \dim(\mathbf{q})}$  (no equality constraint)

```

```

:config-equality-constraint-vector key (update? t) [method]
  return  $\mathbf{b} \in \mathbb{R}^0$  (no equality constraint)

```

```

:square-integration-regular-matrix key (diff-order 1) [method]
(delta-time (/ (- finish-time start-time) 100.0))

```

式 (329) より，関節角微分の二乗積分は次式で得られる．

$$F_{sq,r,k}(\mathbf{p}) = \int_{t_s}^{t_f} \left\| \boldsymbol{\theta}^{(k)}(t) \right\|^2 dt \quad (393)$$

$$= p^T H_{sq r, k} p \quad (394)$$

ただし、

$$H_{sq,r,k} = \int_{t_s}^{t_f} \left(B_{n-k}(t) \hat{D}_k \right)^T B_{n-k}(t) \hat{D}_k dt \quad (395)$$

$$= \int_{t_s}^{t_f} \begin{pmatrix} \left(D^k \mathbf{b}_{n-k}(t) \right) \left(D^k \mathbf{b}_{n-k}(t) \right)^T & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \left(D^k \mathbf{b}_{n-k}(t) \right) \left(D^k \mathbf{b}_{n-k}(t) \right)^T \end{pmatrix} dt \quad (396)$$

これは二次形式の正則化項である.

return $\mathbf{H}_{sq,r,k} \in \mathbb{R}^{dim(\mathbf{p}) \times dim(\mathbf{p})}$

```
:first-differential-square-integration-regular-matrix <key (delta-time (/ (- _finish-time _start-time)
100.0))> [method]
```

return $\mathbf{H}_{sqr,1} \in \mathbb{R}^{dim(\mathbf{p}) \times dim(\mathbf{p})}$

```
:second-differential-square-integration-regular-matrix key (delta-time (/ (- finish-time start-time) 100.0))
```

return $\mathbf{H}_{sqr,2} \in \mathbb{R}^{dim(\mathbf{p}) \times dim(\mathbf{p})}$

```

:third-differential-square-integration-regular-matrix key (delta-time (/ (- finish-time start-time)
100.0))
[method]

```

return $\mathbf{H}_{sqr,3} \in \mathbb{R}^{dim(\mathbf{p}) \times dim(\mathbf{p})}$

```
:control-vector-regular-matrix [method]
```

$$\mathbf{W}_{req,p} := \min(k_{max,p}, \|e\|^2 + k_{off,p})\mathbf{I} + k_{sqr,1}\mathbf{H}_{sqr,1} + k_{sqr,2}\mathbf{H}_{sqr,2} + k_{sqr,3}\mathbf{H}_{sqr,3} \quad (397)$$

return $\mathbf{W}_{req,p} \in \mathbb{R}^{dim(\mathbf{p}) \times dim(\mathbf{p})}$

```
:control-vector-regular-vector [method]
```

$$\mathbf{v}_{reg,p} := (k_{sqr,1} \mathbf{H}_{sqr,1} + k_{sqr,2} \mathbf{H}_{sqr,2} + k_{sqr,3} \mathbf{H}_{sqr,3}) \mathbf{p} \quad (398)$$

return $\mathbf{v}_{reg,p} \in \mathbb{R}^{dim(\mathbf{p})}$

:motion-duration-regular-matrix

[method]

式 (337) より , 動作期間の二乗は次式で得られる .

$$F_{duration}(\mathbf{t}) = |t_1 - t_{N_{tm}}|^2 \quad (399)$$

$$= \mathbf{t}^T \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \mathbf{t} \quad (400)$$

$$= \mathbf{t}^T \mathbf{H}_{duration} \mathbf{t} \quad (401)$$

これは二次形式の正則化項である .

return $\mathbf{H}_{duration} \in \mathbb{R}^{dim(\mathbf{t}) \times dim(\mathbf{t})}$

:timing-vector-regular-matrix

[method]

$$\mathbf{W}_{reg,t} := \min(k_{max,t}, \|\mathbf{e}\|^2 + k_{off,t}) \mathbf{I} + k_{duration} \mathbf{H}_{duration} \quad (402)$$

return $\mathbf{W}_{reg,t} \in \mathbb{R}^{dim(\mathbf{t}) \times dim(\mathbf{t})}$

:timing-vector-regular-vector

[method]

$$\mathbf{v}_{reg,t} := k_{duration} \mathbf{H}_{duration} \mathbf{t} \quad (403)$$

return $\mathbf{v}_{reg,t} \in \mathbb{R}^{dim(\mathbf{t})}$

:regular-matrix

[method]

$$\mathbf{W}_{reg} := \begin{pmatrix} \mathbf{W}_{reg,p} & \\ & \mathbf{W}_{reg,t} \end{pmatrix} \quad (404)$$

return $\mathbf{W}_{reg} \in \mathbb{R}^{dim(\mathbf{q}) \times dim(\mathbf{q})}$

:regular-vector

[method]

$$\mathbf{v}_{reg} := \begin{pmatrix} \mathbf{v}_{reg,p} \\ \mathbf{v}_{reg,t} \end{pmatrix} \quad (405)$$

return $\mathbf{v}_{reg} \in \mathbb{R}^{dim(\mathbf{q}) \times dim(\mathbf{q})}$

:update-collision-inequality-constraint

[method]

Not implemented yet.

:update-viewer *key (trajectory-delta-time (/ (- _finish-time _start-time) 10.0))*

[method]

Update viewer.

:print-status [method]

Print status.

:print-motion-information [method]

Print motion information.

:play-animation $\mathcal{E}key$ (*robot*) [method]

(*delta-time* (/ (- *_finish-time* *_start-time*) 100.0))

(*only-motion-duration?* *t*)

(*loop?* *t*)

(*visualize-callback-func*)

Print motion animation.

:plot-theta-graph $\mathcal{E}key$ (*joint-id* *nil*) [method]

(*divide-num* 200)

(*plot-numerical?* *nil*)

(*only-motion-duration?* *t*)

(*dat-filename* /tmp/bspline-configuration-task-plot-theta-graph.dat)

(*dump-pdf?* *nil*)

(*dump-filename* (ros::resolve-ros-path package://eus-qp/optmotiongen/logs/bspline-configuration-task-plot-theta-graph.pdf))

Plot graph.

:generate-angle-vector-sequence $\mathcal{E}key$ (*divide-num* 100) [method]

Generate angle-vector-sequence.

get-bspline-knot *i n m x_min x_max h* [function]

$$t_i = \frac{i-n}{m-n}(t_f - t_s) + t_s \quad (406)$$

$$= hi + \frac{mt_s - nt_f}{m-n} \quad (407)$$

return knot t_i for B-spline function

bspline-basis-func *x i n m x_min x_max* $\mathcal{E}optional$ (*n-orig* *n*) (*m-orig* *m*) [function]

$$b_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (408)$$

$$b_{i,n}(t) = \frac{(t - t_i)b_{i,n-1}(t) + (t_{i+n+1} - t)b_{i+1,n-1}(t)}{nh} \quad (409)$$

return B-spline function value $b_{i,n}(t)$.

5 補足

5.1 既存のロボット基礎クラスの拡張

joint [class]

```

:super    propertied-object
:slots    (parent-link)
           (child-link)
           (joint-angle)
           (min-angle)
           (max-angle)
           (default-coords)
           (joint-velocity)
           (joint-acceleration)
           (joint-torque)
           (max-joint-velocity)
           (max-joint-torque)
           (joint-min-max-table)
           (joint-min-max-target)

```

:child-link *ℳrest args* [method]

Returns child link of this joint. If any arguments is set, it is passed to the child-link.

Override to support the case that child-link is cascaded-link instantiate. Return the root link of child cascaded-link instantiate in that case.

:axis-vector [method]

Return joint axis vector. Represented in world coordinates.

return $\mathbf{a}_i \in \mathbb{R}^3$

:pos [method]

Return joint position. Represented in world coordinates.

return $\mathbf{p}_i \in \mathbb{R}^3$

bodyset-link [class]

```

:super    bodyset
:slots    (rot)
           (pos)
           (parent)
           (descendants)
           (worldcoords)
           (manager)
           (changed)
           (geometry::bodies)
           (joint)
           (parent-link)
           (child-links)
           (analysis-level)
           (default-coords)
           (weight)
           (acentroid)

```


(inertia-tensor)
 (angular-velocity)
 (angular-acceleration)
 (spacial-velocity)
 (spacial-acceleration)
 (momentum-velocity)
 (angular-momentum-velocity)
 (momentum)
 (angular-momentum)
 (force)
 (moment)
 (ext-force)
 (ext-moment)

:mg [method]
 return $mg = \|m\mathbf{g}\|$

:mg-vec [method]
 return $m\mathbf{g}$

cascaded-link [class]

:super **cascaded-coords**
 :slots (rot)
 (pos)
 (parent)
 (descendants)
 (worldcoords)
 (manager)
 (changed)
 (links)
 (joint-list)
 (bodies)
 (collision-avoidance-links)
 (end-coords-list)

:calc-jacobian-from-joint-list $\mathcal{E}key$ (*union-joint-list*) [method]
(move-target)
(joint-list (mapcar #'(lambda (mt) (send-all (send self :link-list (send mt :parent) (transform-coords (mapcar #'(lambda (mt) (make-coords)) move-target))
(translation-axis (mapcar #'(lambda (mt) t) move-target))
(rotation-axis (mapcar #'(lambda (mt) t) move-target)))

union-joint-list list of all joints considered in jacobian. column num of jacobian is same with length of

union-joint-list.

move-target list of move-target.

joint-list list of joint-list which is contained in each chain of move-target.

transform-coords list of transform-coords of each move-target.

translation-axis list of translation-axis of each move-target.

rotation-axis list of rotation-axis of each move-target.

Get jacobian matrix from following two information: (1) union-joint-list and (2) list of move-target. One recession compared with :calc-jacobian-from-link-list is that child-reverse is not supported. (Only not implemented yet because I do not need such feature in current application.)

:calc-cog-jacobian-from-joint-list *&key (union-joint-list)* [method]
(update-mass-properties t)
(translation-axis :z)

union-joint-list list of all joints considered in jacobian. column num of jacobian is same with length of union-joint-list.

Get CoG jacobian matrix from union-joint-list.

:find-link-route *to &optional from* [method]

Override to support the case that joint does not exist between links. Change from (send to :parent-link) to (send to :parent).

5.2 環境と接触するロボットの関節・リンク構造

2d-planar-contact [class]

| | |
|--------|--|
| :super | cascaded-link |
| :slots | (_contact-coords T_{cnt}) (_contact-pre-coords $T_{cnt-pre}$) |

二次元平面上の長方形領域での接触座標を表す仮想の関節・リンク構造

```

:init  $\mathcal{E}_{key}$  (name contact) [method]
           (contact-pre-offset 100)
    Initialize instance

:contact-coords  $\mathcal{E}_{rest}$  args [method]
    return  $T_{cnt} := \{\mathbf{p}_{cnt}, \mathbf{R}_{cnt}\}$ 

:contact-pre-coords  $\mathcal{E}_{rest}$  args [method]
    return  $T_{cnt-pre} := \{\mathbf{p}_{cnt-pre}, \mathbf{R}_{cnt-pre}\}$ 

:set-from-face  $\mathcal{E}_{key}$  (face) [method]
           (margin 150.0)
    set coords and min/max joint angle from face.

```

robot-environment

[class]

```

:super      cascaded-link
:slots      (_robot  $\mathcal{R}$ )
             (_robot-with-root-virtual  $\hat{\mathcal{R}}$ )
             (_root-virtual-joint-list list of root virtual joint)
             (_contact-list  $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{N_C}\}$ )
             (_variant-joint-list  $\mathcal{J}_{var}$ )
             (_invariant-joint-list  $\mathcal{J}_{invar}$ )
             (_drive-joint-list  $\mathcal{J}_{drive}$ )

```

ロボットとロボット・環境間の接触のクラス．

以下を合わせた関節・リンク構造に関するメソッドが定義されている．

1. 浮遊ルートリンクのための仮想関節付きのロボットの関節
2. 接触位置を定める仮想関節

関節・リンク構造を定めるために，初期化時に以下を与える

robot \mathcal{R} ロボット (cascaded-link クラスのインスタンス) ．

contact-list $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{N_C}\}$ 接触 (2d-planar-contact クラスなどのインスタンス) のリスト ．

ロボット R に，浮遊ルートリンクの変位に対応する仮想関節を付加した仮想関節付きロボット $\hat{\mathcal{R}}$ を内部で保持する ．

```

:init  $\mathcal{E}key$  (robot) [method]
      (contact-list)
      (root-virtual-mode :6dof)
      (root-virtual-joint-class-list)
      (root-virtual-joint-axis-list)
      Initialize instance

:dissoc-root-virtual [method]
      dissoc root virtual parent/child structure.

:init-pose [method]
      set zero joint angle.

:robot  $\mathcal{E}rest$  args [method]
      return  $\mathcal{R}$ 

:robot-with-root-virtual  $\mathcal{E}rest$  args [method]
      return  $\hat{\mathcal{R}}$ 

:contact-list  $\mathcal{E}rest$  args [method]
      return  $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{N_C}\}$ 

:contact name  $\mathcal{E}rest$  args [method]
      return  $\mathcal{C}_i$ 

:variant-joint-list  $\mathcal{E}optional$  (jl :nil) [method]

```

return \mathcal{J}_{var}

:invariant-joint-list *Optional (jl :nil)*

[method]

return \mathcal{J}_{invar}

:drive-joint-list *Optional (jl :nil)*

[method]

return \mathcal{J}_{drive}

:root-virtual-joint-list

[method]

return list of root virtual joint

5.3 irteus の inverse-kinematics 互換関数

cascaded-link

[class]

:super **cascaded-coords**
:slots (rot)
 (pos)
 (parent)
 (descendants)
 (worldcoords)
 (manager)
 (changed)
 (links)
 (joint-list)
 (bodies)
 (collision-avoidance-links)
 (end-coords-list)

:inverse-kinematics-optmotiongen *target-coords* *key* (*stop 50*)

[method]

(*link-list*)

(*move-target*)

(*debug-view*)

(*revert-if-fail t*)

(*transform-coords target-coords*)

(*translation-axis* (*cond* ((*atom move-target*) *t*) (*t* (*make-list*

(*rotation-axis* (*cond* ((*atom move-target*) *t*) (*t* (*make-list*

(*thre* (*cond* ((*atom move-target*) 1) (*t* (*make-list* (*length* *n*

(*rthre* (*cond* ((*atom move-target*) (*deg2rad* 1)) (*t* (*make-list*

(*collision-avoidance-link-pair* :nil)

(*collision-distance-limit* 10.0)

(*obstacles*)

(*min-loop*)

(*root-virtual-mode* :fix)


```

(rotation-axis-list :nil)
(thre 1.0)
(rthre (deg2rad 1))
(thre-list :nil)
(rthre-list :nil)
(collision-avoidance-link-pair :nil)
(collision-distance-limit 10.0)
(obstacles)
(min-loop)
(root-virtual-mode :fix)
(root-virtual-joint-invariant? nil)
(joint-angle-margin 0.0)
(posture-joint-list (make-list (length target-coords) 0))
(posture-joint-angle-list (make-list (length target-coords) 0))
(norm-regular-scale-max 0.001)
(norm-regular-scale-offset 1.000000e-07)
(adjacent-regular-scale 0.0)
(pre-process-func)
(post-process-func)
 $\mathcal{E}$ allow-other-keys

```

Solve inverse kinematics problem with sqp optimization. `target-coords-list` : The coordinate of the target that returns coordinates. Use a list of targets to solve the IK relative to multiple end links simultaneously. Function is not available to `target-coords`. `move-target-list` : Specify end-effector coordinate. When the `target-coords` is list, this should be list too. `stop` : Maximum number for IK iteration. Default is 50. `debug-view` : Set `t` to show debug message and visualization. Use `:no-message` to just show the irtview image. Default is `nil`. `revert-if-fail` : Set `nil` to keep the angle posture of IK solve iteration. Default is `t`, which return to original position when IK fails. `translation-axis-list` : `:x` `:y` `:z` for constraint along the x, y, z axis. `:xy` `:yz` `:zx` for plane. Default is `t`. `rotation-axis-list` : Use `nil` for position only IK. `:x`, `:y`, `:z` for the constraint around axis with plus direction. When the `target-coords` is list, this should be list too. Default is `t`. `thre` : Threshold for position error to terminate IK iteration. Default is 1 [mm]. `rthre` : Threshold for rotation error to terminate IK iteration. Default is 0.017453 [rad] (1 deg).

robot-model

[class]

```

:super      cascaded-link
:slots      (rot)
             (pos)
             (parent)
             (descendants)
             (worldcoords)
             (manager)
             (changed)
             (links)
             (joint-list)
             (bodies)
             (collision-avoidance-links)

```

```

(end-coords-list)
(larm-end-coords)
(rarm-end-coords)
(lleg-end-coords)
(rleg-end-coords)
(head-end-coords)
(torso-end-coords)
(larm-root-link)
(rarm-root-link)
(lleg-root-link)
(rleg-root-link)
(head-root-link)
(torso-root-link)
(larm-collision-avoidance-links)
(rarm-collision-avoidance-links)
(larm)
(rarm)
(lleg)
(rleg)
(torso)
(head)
(force-sensors)
(imu-sensors)
(cameras)
(support-polygons)

```

:limb *limb method ℰrest args* [method]

Extend to support to call :inverse-kinematics-optmotiongen.

contact-ik-arg [class]

```

:super    cascaded-link
:slots    (_contact-coords  $T_{cnt}$ )

```

inverse-kinematics-optmotiongen の *target-coords*, *translation-axis*, *rotation-axis*, *transform-coords* 5 | 数に対応する接触座標を表す仮想の関節・リンク構造 .

:init *ℰkey (target-coords)* [method]
(translation-axis)
(rotation-axis)
(transform-coords)

Initialize instance

:contact-coords *ℰrest args* [method]
 return $T_{cnt} := \{\mathbf{p}_{cnt}, \mathbf{R}_{cnt}\}$

ik-arg-axis->axis-list *ik-arg-axis* [function]
 Convert translation-axis / rotatoin-axis to axis list.

generate-contact-ik-arg-from-rect-face *ℰkey (rect-face)* [function]
(margin (or (send rect-face :get :margin) 0))
 Generate contact-ik-arg instance from rectangle face.

generate-contact-ik-arg-from-line-segment *ℰkey (line-seg)* [function]
(margin (or (send line-seg :get :margin) 0))
 Generate contact-ik-arg instance from line segment.

axis->index *axis* [function]

axis->sgn *axis* [function]

5.4 関節トルク勾配の計算

get-link-jacobian-for-contact-torque *ℰkey (robot)* [function]
(drive-joint-list)
(contact-coords)
(contact-parent-link)

contact-coords に対応する接触部位の番号を m とする．*contact-coords* の位置姿勢を $\mathbf{r}_m \in \mathbb{R}^6$, *drive-joint-list* の関節角度ベクトルを $\psi \in \mathbb{R}^{N_{drive-joint}}$ として，次式を満たすヤコビ行列 \mathbf{J}_m を返す．

$$\dot{\mathbf{r}}_m = \mathbf{J}_m \dot{\psi} \quad (410)$$

$$= \sum_{i=1}^{N_{drive-joint}} \mathbf{j}_m^{(i)} \dot{\psi}_i \quad (411)$$

$$\mathbf{j}_m^{(i)} = \begin{pmatrix} \mathbf{a}_{\psi_i} \times (\mathbf{p}_m - \mathbf{p}_{\psi_i}) \\ \mathbf{a}_{\psi_i} \end{pmatrix} \quad (412)$$

$$\mathbf{J}_m = \begin{pmatrix} \mathbf{j}_m^{(1)} & \mathbf{j}_m^{(2)} & \cdots & \mathbf{j}_m^{N_{drive-joint}} \end{pmatrix} \quad (413)$$

$\mathbf{a}_{\psi_i}, \mathbf{p}_{\psi_i} \in \mathbb{R}^3$ は i 番目の関節の回転軸ベクトルと位置である．

return $\mathbf{J}_m \in \mathbb{R}^{6 \times N_{drive-joint}}$

get-contact-torque *ℰkey (robot)* [function]
(drive-joint-list)
(wrench-list)
(contact-target-coords-list)
(contact-attention-coords-list)

ロボットの接触部位に加わる接触レンチによって生じる関節トルク $\boldsymbol{\tau}^{cnt}$ は，以下で得られる．

$$\boldsymbol{\tau}^{cnt} = \sum_{m=1}^{N_{cnt}} \mathbf{J}_m^T \mathbf{w}_m \quad (414)$$

\mathbf{w}_m は m 番目の接触部位で受ける接触レンチである．

return $\boldsymbol{\tau}^{cnt} \in \mathbb{R}^{N_{drive-joint}}$

get-contact-torque-jacobian *key (robot)* [function]
(joint-list)
(drive-joint-list)
(wrench-list)
(contact-target-coords-list)
(contact-attention-coords-list)

式 (413) の J_m を以下のように分解して利用する .

$$J_m = \begin{pmatrix} j_{x,m}^T \\ j_{y,m}^T \\ j_{z,m}^T \\ j_{R,m}^T \\ j_{P,m}^T \\ j_{Y,m}^T \end{pmatrix} = \begin{pmatrix} \left(\frac{\partial r_{x,m}}{\partial \psi} \right)^T \\ \left(\frac{\partial r_{y,m}}{\partial \psi} \right)^T \\ \left(\frac{\partial r_{z,m}}{\partial \psi} \right)^T \\ \left(\frac{\partial r_{R,m}}{\partial \psi} \right)^T \\ \left(\frac{\partial r_{P,m}}{\partial \psi} \right)^T \\ \left(\frac{\partial r_{Y,m}}{\partial \psi} \right)^T \end{pmatrix} \quad (415)$$

これを式 (414) に代入すると ,

$$\tau^{cnt} = \sum_{m=1}^{N_{cnt}} J_m^T w_m \quad (416)$$

$$= \sum_{m=1}^{N_{cnt}} \begin{pmatrix} j_{x,m} & j_{y,m} & j_{z,m} & j_{R,m} & j_{P,m} & j_{Y,m} \end{pmatrix} \begin{pmatrix} f_{x,m} \\ f_{y,m} \\ f_{z,m} \\ n_{x,m} \\ n_{y,m} \\ n_{z,m} \end{pmatrix} \quad (417)$$

$$= \sum_{m=1}^{N_{cnt}} (j_{x,m} f_{x,m} + j_{y,m} f_{y,m} + j_{z,m} f_{z,m} + j_{R,m} n_{x,m} + j_{P,m} n_{y,m} + j_{Y,m} n_{z,m}) \quad (418)$$

$$= \sum_{m=1}^{N_{cnt}} \left(\frac{\partial r_{x,m}}{\partial \psi} f_{x,m} + \frac{\partial r_{y,m}}{\partial \psi} f_{y,m} + \frac{\partial r_{z,m}}{\partial \psi} f_{z,m} + \frac{\partial r_{R,m}}{\partial \psi} n_{x,m} + \frac{\partial r_{P,m}}{\partial \psi} n_{y,m} + \frac{\partial r_{Y,m}}{\partial \psi} n_{z,m} \right) \quad (419)$$

joint-list の関節角度ベクトルを $\theta \in \mathbb{R}^{N_{joint}}$, *drive-joint-list* の関節角度ベクトルを $\psi \in \mathbb{R}^{N_{drive-joint}}$ とする . トルク勾配行列 $\frac{\partial \tau^{cnt}}{\partial \theta}$ は次式で得られる .

$$\frac{\partial \tau^{cnt}}{\partial \theta} = \sum_{m=1}^{N_{cnt}} \left(\frac{\partial J_m}{\partial \theta} \right)^T w_m \quad (420)$$

$$= \sum_{m=1}^{N_{cnt}} \left(f_{x,m} \frac{\partial^2 r_{x,m}}{\partial \psi \partial \theta} + f_{y,m} \frac{\partial^2 r_{y,m}}{\partial \psi \partial \theta} + f_{z,m} \frac{\partial^2 r_{z,m}}{\partial \psi \partial \theta} + n_{x,m} \frac{\partial^2 r_{R,m}}{\partial \psi \partial \theta} + n_{y,m} \frac{\partial^2 r_{P,m}}{\partial \psi \partial \theta} + n_{z,m} \frac{\partial^2 r_{Y,m}}{\partial \psi \partial \theta} \right) \begin{bmatrix} \frac{\partial^2 r_{x,m}}{\partial \psi_i \partial \theta_j} \\ \frac{\partial^2 r_{y,m}}{\partial \psi_i \partial \theta_j} \\ \frac{\partial^2 r_{z,m}}{\partial \psi_i \partial \theta_j} \\ \frac{\partial^2 r_{R,m}}{\partial \psi_i \partial \theta_j} \\ \frac{\partial^2 r_{P,m}}{\partial \psi_i \partial \theta_j} \\ \frac{\partial^2 r_{Y,m}}{\partial \psi_i \partial \theta_j} \end{bmatrix}_{i=1, \dots, N_{drive-joint}, j=1, \dots, N_{joint}} \quad (422)$$

return $\frac{\partial \tau^{cnt}}{\partial \theta} \in \mathbb{R}^{N_{drive-joint} \times N_{joint}}$

get-link-jacobian-for-gravity-torque *ℰkey (robot)* [function]
(drive-joint-list)
(gravity-link)

gravity-link のリンク番号を k とする．*gravity-link* の重心位置を $\mathbf{p}_{cog,k} \in \mathbb{R}^3$, *drive-joint-list* の関節角度ベクトルを $\boldsymbol{\psi} \in \mathbb{R}^{N_{drive-joint}}$ として，次式を満たすヤコビ行列 $\mathbf{J}_{cog,k}$ を返す．

$$\dot{\mathbf{p}}_{cog,k} = \mathbf{J}_{cog,k} \dot{\boldsymbol{\psi}} \quad (423)$$

$$= \sum_{i=1}^{N_k} \mathbf{j}_{cog,k}^{(i)} \dot{\psi}_i \quad (424)$$

$$\mathbf{j}_{cog,k}^{(i)} = \begin{cases} \mathbf{a}_{\psi_i} \times (\mathbf{p}_{cog,k} - \mathbf{p}_{\psi_i}) & (1 \leq i \leq N_k) \\ \mathbf{0}_3 & (N_k + 1 \leq i \leq N_{drive-joint}) \end{cases} \quad (425)$$

$\mathbf{a}_{\psi_i}, \mathbf{p}_{\psi_i} \in \mathbb{R}^3$ は i 番目の関節の回転軸ベクトルと位置である．*gravity-link* よりもルート側にある関節の番号を $1, \dots, N_k$, *gravity-link* よりも末端側にある関節の番号を $N_k + 1, \dots, N_{drive-joint}$ とする．リンクの重心位置と関節角度の依存関係から，ヤコビ行列の右には次式のように零ベクトルが並ぶ．

$$\mathbf{J}_{cog,k} = \begin{pmatrix} \mathbf{j}_{cog,k}^{(1)} & \cdots & \mathbf{j}_{cog,k}^{(N_k)} & \mathbf{j}_{cog,k}^{(N_k+1)} & \cdots & \mathbf{j}_{cog,k}^{(N_{drive-joint})} \end{pmatrix} \quad (426)$$

$$= \begin{pmatrix} \mathbf{j}_{cog,k}^{(1)} & \cdots & \mathbf{j}_{cog,k}^{(N_k)} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \quad (427)$$

return $\mathbf{J}_{cog,k} \in \mathbb{R}^{3 \times N_{drive-joint}}$

get-gravity-torque *ℰkey (robot)* [function]
(drive-joint-list)
(gravity-link-list)

ロボットのリンク自重によって生じる関節トルク $\boldsymbol{\tau}^{grav}$ は，ロボットモーション P111 式 (3.3.22) より以下で得られる．

$$\boldsymbol{\tau}^{grav} = \left(\sum_{k=1}^{N_{gravity-link}} m_k \mathbf{J}_{cog,k}^T \right) \mathbf{g} \quad (428)$$

m_k は k 番目のリンクの質量である．

return $\boldsymbol{\tau}^{grav} \in \mathbb{R}^{N_{drive-joint}}$

get-gravity-torque-jacobian *ℰkey (robot)* [function]
(joint-list)
(drive-joint-list)
(gravity-link-list)

式 (427) の $\mathbf{J}_{cog,k}$ を以下のように分解して利用する．

$$\mathbf{J}_{cog,k} = \begin{pmatrix} \mathbf{j}_{cog,x,k}^T \\ \mathbf{j}_{cog,y,k}^T \\ \mathbf{j}_{cog,z,k}^T \end{pmatrix} = \begin{pmatrix} \left(\frac{\partial \mathbf{p}_{cog,x,k}}{\partial \boldsymbol{\psi}} \right)^T \\ \left(\frac{\partial \mathbf{p}_{cog,y,k}}{\partial \boldsymbol{\psi}} \right)^T \\ \left(\frac{\partial \mathbf{p}_{cog,z,k}}{\partial \boldsymbol{\psi}} \right)^T \end{pmatrix} \quad (429)$$

これを式 (428) に代入すると ,

$$\boldsymbol{\tau}^{grav} = \left(\sum_{k=1}^{N_{gravity-link}} m_k \mathbf{J}_{cog,k}^T \right) \mathbf{g} \quad (430)$$

$$= \sum_{k=1}^{N_{gravity-link}} m_k \begin{pmatrix} \dot{\mathbf{j}}_{cog,x,k} & \dot{\mathbf{j}}_{cog,y,k} & \dot{\mathbf{j}}_{cog,z,k} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (431)$$

$$= \sum_{k=1}^{N_{gravity-link}} m_k g \dot{\mathbf{j}}_{cog,z,k} \quad (432)$$

$$= \sum_{k=1}^{N_{gravity-link}} m_k g \frac{\partial p_{cog,z,k}}{\partial \psi} \quad (433)$$

joint-list の関節角度ベクトルを $\boldsymbol{\theta} \in \mathbb{R}^{N_{joint}}$, *drive-joint-list* の関節角度ベクトルを $\boldsymbol{\psi} \in \mathbb{R}^{N_{drive-joint}}$ とする . トルク勾配行列 $\frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}}$ は次式で得られる . これは対称行列である .

$$\frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}} = \sum_{k=1}^{N_{gravity-link}} m_k g \frac{\partial^2 p_{cog,z,k}}{\partial \boldsymbol{\psi} \partial \boldsymbol{\theta}} \quad (434)$$

$$\frac{\partial^2 p_{cog,z,k}}{\partial \boldsymbol{\psi} \partial \boldsymbol{\theta}} = \left[\frac{\partial^2 p_{cog,z,k}}{\partial \psi_i \partial \theta_j} \right]_{i=1, \dots, N_{drive-joint}, j=1, \dots, N_{joint}} \quad (435)$$

つまり

$$\frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}} = \left[\sum_{k=1}^{N_{gravity-link}} m_k g \frac{\partial^2 p_{cog,z,k}}{\partial \psi_i \partial \theta_j} \right]_{i=1, \dots, N_{drive-joint}, j=1, \dots, N_{joint}} \quad (436)$$

return $\frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{N_{drive-joint} \times N_{joint}}$