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 32 年 2 月 25 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　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

$\boldsymbol{q} \in \mathbb{R}^{n_q}$ 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，$\boldsymbol{q}$ 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 $\boldsymbol{e}(\boldsymbol{q}) : \mathbb{R}^{n_q} \to \mathbb{R}^{n_e}$ について，次式を満たす $\boldsymbol{q}$ を得ることとして定義する．

$$\boldsymbol{e}(\boldsymbol{q}) = \boldsymbol{0} \tag{1.1}$$

例えば一般の逆運動学計算では，$\boldsymbol{e}(\boldsymbol{q})$ 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 (1.1) 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 (1.1) 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 (1.1) の求解は次の最適化問題として表される [1]．

$$\min_{\boldsymbol{q}} \ F(\boldsymbol{q}) \tag{1.2a}$$

$$\text{where} \ \ F(\boldsymbol{q}) \overset{\text{def}}{=} \frac{1}{2}\|\boldsymbol{e}(\boldsymbol{q})\|^2 \tag{1.2b}$$

コンフィギュレーションが最小値 $\boldsymbol{q}_{min}$ と最大値 $\boldsymbol{q}_{max}$ 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．

$$\min_{\boldsymbol{q}} \ F(\boldsymbol{q}) \quad \text{s.t.} \ \ \boldsymbol{q}_{min} \leq \boldsymbol{q} \leq \boldsymbol{q}_{max} \tag{1.3}$$

例えば一般の逆運動学計算では，$\boldsymbol{q}_{min}, \boldsymbol{q}_{max}$ 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 (1.3) 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]．

$$\min_{\boldsymbol{q}} \ F(\boldsymbol{q}) \tag{1.4a}$$

$$\text{s.t.} \ \ \boldsymbol{A}\boldsymbol{q} = \bar{\boldsymbol{b}} \tag{1.4b}$$

$$\boldsymbol{C}\boldsymbol{q} \geq \bar{\boldsymbol{d}} \tag{1.4c}$$

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 $\Delta\boldsymbol{q}_k^*$ を用いて，$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \Delta\boldsymbol{q}_k^*$ として反復更新することで，式 (1.4) の最適解を導出する [3]．

$$\min_{\Delta\boldsymbol{q}_k} \ F(\boldsymbol{q}_k) + \nabla F(\boldsymbol{q}_k)^T \Delta\boldsymbol{q}_k + \frac{1}{2}\Delta\boldsymbol{q}_k^T \nabla^2 F(\boldsymbol{q}_k) \Delta\boldsymbol{q}_k \tag{1.5a}$$

$$\text{s.t.} \ \ \boldsymbol{A}\Delta\boldsymbol{q}_k = \bar{\boldsymbol{b}} - \boldsymbol{A}\boldsymbol{q}_k \tag{1.5b}$$

$$\boldsymbol{C}\Delta\boldsymbol{q}_k \geq \bar{\boldsymbol{d}} - \boldsymbol{C}\boldsymbol{q}_k \tag{1.5c}$$

[1] 任意の半正定値行列 $\boldsymbol{W}$ に対して，$\|\boldsymbol{e}(\boldsymbol{q})\|_{\boldsymbol{W}}^2 = \boldsymbol{e}(\boldsymbol{q})^T \boldsymbol{W} \boldsymbol{e}(\boldsymbol{q}) = \boldsymbol{e}(\boldsymbol{q})^T \boldsymbol{S}^T \boldsymbol{S} \boldsymbol{e}(\boldsymbol{q}) = \|\boldsymbol{S}\boldsymbol{e}(\boldsymbol{q})\|^2$ を満たす $\boldsymbol{S}$ が必ず存在するので，式 (1.2b) は任意の重み付きノルムを表現可能である．

[2] 式 (1.3) 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．

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

[3] 式 (1.5a) は $F(\boldsymbol{q})$ を $\boldsymbol{q}_k$ 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`.

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

$$
\begin{aligned}
\nabla F(\boldsymbol{q}) &= \left(\frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T \boldsymbol{e}(\boldsymbol{q}) && (1.6a)\\
&= \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{e}(\boldsymbol{q}) && (1.6b)\\
\nabla^2 F(\boldsymbol{q}) &= \sum_{i=1}^{m} e_i(\boldsymbol{q}) \nabla^2 e_i(\boldsymbol{q}) + \left(\frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T \frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}} && (1.6c)\\
&\approx \left(\frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T \frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}} && (1.6d)\\
&= \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{J}(\boldsymbol{q}) && (1.6e)
\end{aligned}
$$

ただし，$e_i(\boldsymbol{q}) \ \ (i = 1, 2, \cdots, m)$ は $\boldsymbol{e}(\boldsymbol{q})$ の i 番目の要素である．式 (1.6c) から式 (1.6d) への変形では $\boldsymbol{e}(\boldsymbol{q})$ の二階微分がゼロであると近似している．$\boldsymbol{J}(\boldsymbol{q}) \overset{\text{def}}{=} \frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{n_e \times n_q}$ は $\boldsymbol{e}(\boldsymbol{q})$ のヤコビ行列である．

式 (1.6a) , 式 (1.6d) から式 (1.5a) の目的関数は次式で表される [5]．

$$
\begin{aligned}
& \frac{1}{2}\boldsymbol{e}_k^T \boldsymbol{e}_k + \boldsymbol{e}_k^T \boldsymbol{J}_k \Delta \boldsymbol{q}_k + \frac{1}{2} \Delta \boldsymbol{q}_k^T \boldsymbol{J}_k^T \boldsymbol{J}_k \Delta \boldsymbol{q}_k && (1.7a)\\
=\ & \frac{1}{2} \|\boldsymbol{e}_k + \boldsymbol{J}_k \Delta \boldsymbol{q}_k\|^2 && (1.7b)
\end{aligned}
$$

ただし，$\boldsymbol{e}_k \overset{\text{def}}{=} \boldsymbol{e}(\boldsymbol{q}_k), \boldsymbol{J}_k \overset{\text{def}}{=} \boldsymbol{J}(\boldsymbol{q}_k)$ 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 (1.5) は次式で表される．

$$
\begin{aligned}
& \min_{\Delta \boldsymbol{q}_k} \ \frac{1}{2} \Delta \boldsymbol{q}_k^T \boldsymbol{J}_k^T \boldsymbol{J}_k \Delta \boldsymbol{q}_k + \boldsymbol{e}_k^T \boldsymbol{J}_k \Delta \boldsymbol{q}_k && (1.8a)\\
& \text{s.t.} \ \ \boldsymbol{A} \Delta \boldsymbol{q}_k = \boldsymbol{b} && (1.8b)\\
& \qquad \boldsymbol{C} \Delta \boldsymbol{q}_k \geq \boldsymbol{d} && (1.8c)
\end{aligned}
$$

ここで，

$$
\begin{aligned}
\boldsymbol{b} &= \bar{\boldsymbol{b}} - \boldsymbol{A}\boldsymbol{q}_k && (1.9)\\
\boldsymbol{d} &= \bar{\boldsymbol{d}} - \boldsymbol{C}\boldsymbol{q}_k && (1.10)
\end{aligned}
$$

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 (1.2a) の最適化問題の目的関数を，次式の $\hat{F}(\boldsymbol{q})$ で置き換える．

$$
\begin{aligned}
& \hat{F}(\boldsymbol{q}) = F(\boldsymbol{q}) + F_{reg}(\boldsymbol{q}) && (1.11)\\
& \text{where} \ \ F_{reg}(\boldsymbol{q}) = \frac{1}{2} \boldsymbol{q}^T \bar{\boldsymbol{W}}_{reg} \boldsymbol{q} && (1.12)
\end{aligned}
$$

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

$$
\begin{aligned}
\nabla \hat{F}(\boldsymbol{q}) &= \nabla F(\boldsymbol{q}) + \nabla F_{reg}(\boldsymbol{q}) && (1.13a)\\
&= \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{e}(\boldsymbol{q}) + \bar{\boldsymbol{W}}_{reg} \boldsymbol{q} && (1.13b)\\
\nabla^2 \hat{F}(\boldsymbol{q}) &= \nabla^2 F(\boldsymbol{q}) + \nabla^2 F_{reg}(\boldsymbol{q}) && (1.13c)\\
&\approx \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{J}(\boldsymbol{q}) + \bar{\boldsymbol{W}}_{reg} && (1.13d)
\end{aligned}
$$

[4] 式 (1.5a) の $\nabla^2 F(\boldsymbol{q}_k)$ の部分は一般にはラグランジュ関数の $\boldsymbol{q}_k$ 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 $F(\boldsymbol{q}_k)$ のヘッセ行列と等価になる．

[5] 式 (1.7b) 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 (1.8) の二次計画問題は次式で表される．

$$\min_{\Delta \boldsymbol{q}_k} \ \frac{1}{2}\Delta \boldsymbol{q}_k^T \left(\boldsymbol{J}_k^T \boldsymbol{J}_k + \bar{\boldsymbol{W}}_{reg}\right) \Delta \boldsymbol{q}_k + \left(\boldsymbol{J}_k^T \boldsymbol{e}_k + \bar{\boldsymbol{W}}_{reg} \boldsymbol{q}_k\right)^T \Delta \boldsymbol{q}_k \tag{1.14a}$$

$$\text{s.t.} \ \ \boldsymbol{A}\Delta \boldsymbol{q}_k = \boldsymbol{b} \tag{1.14b}$$

$$\boldsymbol{C}\Delta \boldsymbol{q}_k \geq \boldsymbol{d} \tag{1.14c}$$

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 (1.14a) 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]．

$$\min_{\Delta \boldsymbol{q}_k} \ \frac{1}{2}\Delta \boldsymbol{q}_k^T \left(\boldsymbol{J}_k^T \boldsymbol{J}_k + \bar{\boldsymbol{W}}_{reg} + \lambda \boldsymbol{I}\right) \Delta \boldsymbol{q}_k + \left(\boldsymbol{J}_k^T \boldsymbol{e}_k + \bar{\boldsymbol{W}}_{reg} \boldsymbol{q}_k\right)^T \Delta \boldsymbol{q}_k \tag{1.15a}$$

$$\text{s.t.} \ \ \boldsymbol{A}\Delta \boldsymbol{q}_k = \boldsymbol{b} \tag{1.15b}$$

$$\boldsymbol{C}\Delta \boldsymbol{q}_k \geq \boldsymbol{d} \tag{1.15c}$$

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．

$$\lambda = \lambda_r F(\boldsymbol{q}_k) + w_r \tag{1.16}$$

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

$$\boldsymbol{W}_{reg} \stackrel{\text{def}}{=} \bar{\boldsymbol{W}}_{reg} + \lambda \boldsymbol{I} \tag{1.17a}$$

$$\boldsymbol{v}_{reg} \stackrel{\text{def}}{=} \bar{\boldsymbol{W}}_{reg} \boldsymbol{q}_k \tag{1.17b}$$

とすると，式 (1.15) は次式で表される．

$$\min_{\Delta \boldsymbol{q}_k} \ \frac{1}{2}\Delta \boldsymbol{q}_k^T \left(\boldsymbol{J}_k^T \boldsymbol{J}_k + \boldsymbol{W}\right) \Delta \boldsymbol{q}_k + \left(\boldsymbol{J}_k^T \boldsymbol{e}_k + \boldsymbol{v}_{reg}\right)^T \Delta \boldsymbol{q}_k \tag{1.18a}$$

$$\text{s.t.} \ \ \boldsymbol{A}\Delta \boldsymbol{q}_k = \boldsymbol{b} \tag{1.18b}$$

$$\boldsymbol{C}\Delta \boldsymbol{q}_k \geq \boldsymbol{d} \tag{1.18c}$$

第 2 節や第 4 章で説明する ****-configuration-task* クラスのメソッドは式 (1.18) 中の記号と以下のように対応している．

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

[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 $\boldsymbol{q}$ の取得・更新，タスク関数 $\boldsymbol{e}(\boldsymbol{q})$ の取得，タスク関数のヤコビ行列 $\boldsymbol{J}(\boldsymbol{q}) \overset{\mathrm{def}}{=} \frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ の取得，コンフィギュレーションの等式・不等式制約 $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{C}, \boldsymbol{d}$ 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 $\boldsymbol{q}$ が瞬時の情報，第 2.2 節ではコンフィギュレーション $\boldsymbol{q}$ 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 $\boldsymbol{\theta}$ [rad] [m])
(_wrench-vector $\hat{\boldsymbol{w}}$ [N] [Nm])
(_torque-vector $\boldsymbol{\tau}$ [Nm])
(_phi-vector $\boldsymbol{\phi}$ [rad] [m])
(_num-kin $N_{kin} := |\mathcal{T}^{kin\text{-}trg}| = |\mathcal{T}^{kin\text{-}att}|$)
(_num-contact $N_{cnt} := |\mathcal{T}^{cnt\text{-}trg}| = |\mathcal{T}^{cnt\text{-}att}|$)
(_num-variant-joint $N_{var\text{-}joint} := |\mathcal{J}_{var}|$)
(_num-invariant-joint $N_{invar\text{-}joint} := |\mathcal{J}_{invar}|$)
(_num-drive-joint $N_{drive\text{-}joint} := |\mathcal{J}_{drive}|$)
(_num-posture-joint $N_{posture\text{-}joint} := |\mathcal{J}_{posture}|$)
(_num-external N_{ex} := number of external wrenches)
(_num-collision N_{col} := number of collision check pairs)
(_dim-theta $dim(\boldsymbol{\theta}) = N_{var\text{-}joint}$)
(_dim-wrench $dim(\hat{\boldsymbol{w}}) = 6N_{cnt}$)
(_dim-torque $dim(\boldsymbol{\tau}) = N_{drive\text{-}joint}$)
(_dim-phi $dim(\boldsymbol{\phi}) = N_{invar\text{-}joint}$)
(_dim-variant-config $dim(\boldsymbol{q}_{var})$)
(_dim-invariant-config $dim(\boldsymbol{q}_{invar})$)
(_dim-config $dim(\boldsymbol{q})$)
(_dim-task $dim(\boldsymbol{e})$)
(_kin-scale-mat-list K_{kin})
(_target-posture-scale-list $k_{posture}$)

(_norm-regular-scale-max k_{max})
(_norm-regular-scale-coeff k_{coeff})
(_norm-regular-scale-offset k_{off})
(_torque-regular-scale k_{trq})
(_wrench-maximize-scale $k_{w\text{-}max}$)
(_variant-joint-list $\mathcal{J}_{var}$)
(_invariant-joint-list $\mathcal{J}_{invar}$)
(_drive-joint-list $\mathcal{J}_{drive}$)
(_kin-target-coords-list $\mathcal{T}^{kin\text{-}trg}$)
(_kin-attention-coords-list $\mathcal{T}^{kin\text{-}att}$)
(_contact-target-coords-list $\mathcal{T}^{cnt\text{-}trg}$)
(_contact-attention-coords-list $\mathcal{T}^{cnt\text{-}att}$)
(_variant-joint-angle-margin margin of $\boldsymbol{\theta}$ [deg] [mm])
(_invariant-joint-angle-margin margin of $\boldsymbol{\phi}$ [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 $\bar{\boldsymbol{\theta}}^{trg}$)
(_external-wrench-list $\{\boldsymbol{w}_1^{ex}, \boldsymbol{w}_2^{ex}, \cdots, \boldsymbol{w}_{N_{ex}}^{ex}\}$)
(_external-coords-list $\{T_1^{ex}, T_2^{ex}, \cdots, T_{N_{ex}}^{ex}\}$)
(_wrench-maximize-direction-list $\{\boldsymbol{d}_1^{w\text{-}max}, \boldsymbol{d}_2^{w\text{-}max}, \cdots, \boldsymbol{d}_{N_{cnt}}^{w\text{-}max}\}$)
(_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 $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{var}}$)
(_invariant-task-jacobi buffer for $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{invar}}$)
(_task-jacobi buffer for $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}}$)
(_collision-theta-inequality-constraint-matrix buffer for $\boldsymbol{C}_{col,\theta}$)
(_collision-phi-inequality-constraint-matrix buffer for $\boldsymbol{C}_{col,\phi}$)
(_collision-inequality-constraint-vector buffer for $\boldsymbol{C}_{col}$)

瞬時コンフィギュレーション $\boldsymbol{q}^{(l)}$ と瞬時タスク関数 $\boldsymbol{e}^{(l)}(\boldsymbol{q}^{(l)})$ 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 $\boldsymbol{x}^{(l)}$ と表されるべきだが，このクラス内の説明では省略して $\boldsymbol{x}$ 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 $\boldsymbol{q}$ の取得・更新，タスク関数 $\boldsymbol{e}(\boldsymbol{q})$ の取得，タスク関数のヤコビ行列 $\frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ の取得，コンフィギュレーションの等式・不等式制約 $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{C}, \boldsymbol{d}$ 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 $\mathcal{T}^{kin\text{-}trg}$ 幾何到達目標位置姿勢リスト

  kin-attention-coords-list $\mathcal{T}^{kin\text{-}att}$ 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 $\mathcal{T}^{cnt\text{-}trg}$ 接触目標位置姿勢リスト

  contact-attention-coords-list $\mathcal{T}^{cnt\text{-}att}$ 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 $\bar{\boldsymbol{\theta}}^{trg}$ 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)

- 接触力最大化 (必要な場合のみ)

  wrench-maximize-direction-list 接触レンチ最大化方向 (ワールド座標系で表す)

- 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-coeff k_{coeff} コンフィギュレーション更新量正則化の係数

  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

  wrench-maximize-scale $k_{w\text{-}max}$ 接触レンチ最大化の重み

コンフィギュレーション $\boldsymbol{q}$ は以下から構成される．

$$\boldsymbol{q} := \begin{pmatrix} \boldsymbol{\theta}^T & \hat{\boldsymbol{w}}^T & \boldsymbol{\tau}^T & \boldsymbol{\phi}^T \end{pmatrix}^T \tag{2.1}$$

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

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

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

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

$\hat{\boldsymbol{w}}$ 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．

$$\hat{\boldsymbol{w}} = \begin{pmatrix} \boldsymbol{w}_1^T & \boldsymbol{w}_2^T & \cdots & \boldsymbol{w}_{N_{cnt}}^T \end{pmatrix}^T \tag{2.2}$$

$$= \begin{pmatrix} \boldsymbol{f}_1^T & \boldsymbol{n}_1^T & \boldsymbol{f}_2^T & \boldsymbol{n}_2^T & \cdots & \boldsymbol{f}_{N_{cnt}}^T & \boldsymbol{n}_{N_{cnt}}^T \end{pmatrix}^T \tag{2.3}$$

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

$$\boldsymbol{e}(\boldsymbol{q}) := \begin{pmatrix} \boldsymbol{e}^{kinT}(\boldsymbol{q}) & \boldsymbol{e}^{eom\text{-}transT}(\boldsymbol{q}) & \boldsymbol{e}^{eom\text{-}rotT}(\boldsymbol{q}) & \boldsymbol{e}^{trqT}(\boldsymbol{q}) & \boldsymbol{e}^{postureT}(\boldsymbol{q}) \end{pmatrix}^T \tag{2.4}$$

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

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

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

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

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

:init *&key* *(name)* [method]
(robot-env)
(variant-joint-list (send robot-env :variant-joint-list))
(invariant-joint-list (send robot-env :invariant-joint-list))
(drive-joint-list (send robot-env :drive-joint-list))
(only-kinematics?)
(kin-target-coords-list)
(kin-attention-coords-list)
(contact-target-coords-list)
(contact-attention-coords-list)
(variant-joint-angle-margin 3.0)
(invariant-joint-angle-margin 3.0)
(delta-linear-joint)
(delta-rotational-joint)
(contact-constraint-list (send-all contact-attention-coords-list :get :contact-constraint))
(posture-joint-list)
(posture-joint-angle-list)
(external-wrench-list)
(external-coords-list)
(wrench-maximize-direction-list)
(collision-pair-list)
(collision-distance-margin 0.01)
(collision-distance-margin-list)
(kin-scale 1.0)
(kin-scale-list)
(kin-scale-mat-list)
(target-posture-scale 0.001)
(target-posture-scale-list)
(norm-regular-scale-max (if only-kinematics? 0.001 1.000000e-05))
(norm-regular-scale-coeff 1.0)
(norm-regular-scale-offset 1.000000e-07)
(torque-regular-scale 1.000000e-04)
(wrench-maximize-scale 0)
&allow-other-keys

Initialize instance

:robot-env [method]

return robot-environment instance

:variant-joint-list [method]

return $\mathcal{J}_{var}$

:invariant-joint-list [method]

return $\mathcal{J}_{invar}$

:drive-joint-list [method]

return $\mathcal{J}_{drive}$

:only-kinematics? [method]

return whether to consider only kinematics or not

: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 $N_{kin} := |\mathcal{T}^{kin\text{-}trg}| = |\mathcal{T}^{kin\text{-}att}|$

:num-contact [method]

return $N_{cnt} := |\mathcal{T}^{cnt\text{-}trg}| = |\mathcal{T}^{cnt\text{-}att}|$

:num-variant-joint [method]

return $N_{var\text{-}joint} := |\mathcal{J}_{var}|$

:num-invariant-joint [method]

return $N_{invar\text{-}joint} := |\mathcal{J}_{invar}|$

:num-drive-joint [method]

return $N_{drive\text{-}joint} := |\mathcal{J}_{drive}|$

:num-posture-joint [method]

return $N_{target\text{-}joint} := |\mathcal{J}_{target}|$

: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]

$$\begin{aligned} dim(\boldsymbol{q_{var}}) &:= dim(\boldsymbol{\theta}) + dim(\hat{\boldsymbol{w}}) + dim(\boldsymbol{\tau}) && (2.5) \\ &= N_{var\text{-}joint} + 6N_{cnt} + N_{drive\text{-}joint} && (2.6) \end{aligned}$$

return $dim(\boldsymbol{q_{var}})$

:dim-invariant-config [method]

return $dim(\boldsymbol{q_{invar}}) := dim(\boldsymbol{\phi}) = N_{invar\text{-}joint}$

:dim-config [method]

return $dim(\boldsymbol{q}) := dim(\boldsymbol{q_{var}}) + dim(\boldsymbol{q_{invar}})$

:dim-task [method]

$$\begin{aligned} dim(\boldsymbol{e}) &:= dim(\boldsymbol{e}^{kin}) + dim(\boldsymbol{e}^{eom\text{-}trans}) + dim(\boldsymbol{e}^{eom\text{-}rot}) + dim(\boldsymbol{e}^{trq}) + dim(\boldsymbol{e}^{posture}) & (2.7) \\ &= 6N_{kin} + 3 + 3 + N_{drive\text{-}joint} + N_{posture\text{-}joint} & (2.8) \end{aligned}$$

return $dim(\boldsymbol{e})$

:variant-config-vector [method]

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

:invariant-config-vector [method]

return $\boldsymbol{q_{invar}} := \boldsymbol{\phi}$

:config-vector [method]

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

:set-theta *theta-new &key (relative? nil)*
(apply-to-robot? t) [method]

Set $\boldsymbol{\theta}$.

:set-wrench *wrench-new &key (relative? nil)* [method]

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

:set-torque *torque-new &key (relative? nil)* [method]

Set $\boldsymbol{\tau}$.

:set-phi *phi-new &key (relative? nil)*
(apply-to-robot? t) [method]

Set $\boldsymbol{\phi}$.

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

Set $\boldsymbol{q_{var}}$.

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

Set $\boldsymbol{q_{invar}}$.

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

Set $\boldsymbol{q}$.

:kin-target-coords-list [method]

$$T_m^{kin\text{-}trg} = \{\boldsymbol{p}_m^{kin\text{-}trg}, \boldsymbol{R}_m^{kin\text{-}trg}\} \quad (m = 1, 2, \cdots, N_{kin}) \tag{2.9}$$

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

:kin-attention-coords-list [method]

$$T_m^{kin\text{-}att} = \{\boldsymbol{p}_m^{kin\text{-}att}, \boldsymbol{R}_m^{kin\text{-}att}\} \quad (m = 1, 2, \cdots, N_{kin}) \tag{2.10}$$

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

:contact-target-coords-list [method]

$$T_m^{cnt\text{-}trg} = \{\boldsymbol{p}_m^{cnt\text{-}trg}, \boldsymbol{R}_m^{cnt\text{-}trg}\} \quad (m = 1, 2, \cdots, N_{cnt}) \tag{2.11}$$

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

:contact-attention-coords-list [method]

$$T_m^{cnt\text{-}att} = \{\boldsymbol{p}_m^{cnt\text{-}att}, \boldsymbol{R}_m^{cnt\text{-}att}\} \quad (m = 1, 2, \cdots, N_{cnt}) \tag{2.12}$$

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

:contact-constraint-list [method]

return list of contact-constraint instance

:wrench-list [method]

return $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{N_{cnt}}\}$

:force-list [method]

return $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{N_{cnt}}\}$

:moment-list [method]

return $\{\boldsymbol{n}_1, \boldsymbol{n}_2, \cdots, \boldsymbol{n}_{N_{cnt}}\}$

:external-wrench-list *&optional (new-external-wrench-list :nil)* [method]

set / get $\{\boldsymbol{w}_1^{ex}, \boldsymbol{w}_2^{ex}, \cdots, \boldsymbol{w}_{N_{ex}}^{ex}\}$

:external-force-list [method]

return $\{\boldsymbol{f}_1^{ex}, \boldsymbol{f}_2^{ex}, \cdots, \boldsymbol{f}_{N_{ex}}^{ex}\}$

:external-moment-list [method]

return $\{\boldsymbol{n}_1^{ex}, \boldsymbol{n}_2^{ex}, \cdots, \boldsymbol{n}_{N_{ex}}^{ex}\}$

:mg-vec [method]

return $m\boldsymbol{g}$

:cog *&key (update? t)* [method]

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

:kinematics-task-value *&key (update? t)* [method]

$$
\begin{aligned}
\boldsymbol{e}^{kin}(\boldsymbol{q}) &= \boldsymbol{e}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) && (2.13)\\
&= \begin{pmatrix} \boldsymbol{e}_1^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \boldsymbol{e}_2^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \vdots \\ \boldsymbol{e}_{N_{kin}}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \end{pmatrix} && (2.14)\\
\boldsymbol{e}_m^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= K_{kin} \begin{pmatrix} \boldsymbol{p}_m^{kin\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}_m^{kin\text{-}att}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \boldsymbol{a}\left(\boldsymbol{R}_m^{kin\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) \boldsymbol{R}_m^{kin\text{-}att}(\boldsymbol{\theta}, \boldsymbol{\phi})^T\right) \end{pmatrix} \in \mathbb{R}^6 \quad (m = 1, 2, \cdots, N_{kin}) && (2.15)
\end{aligned}
$$

$\boldsymbol{a}(\boldsymbol{R})$ は姿勢行列 $\boldsymbol{R}$ の等価角軸ベクトルを表す.

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

:eom-trans-task-value *&key (update? t)* [method]

$$
\begin{aligned}
\boldsymbol{e}^{eom\text{-}trans}(\boldsymbol{q}) &= \boldsymbol{e}^{eom\text{-}trans}(\hat{\boldsymbol{w}}) && (2.16)\\
&= \sum_{m=1}^{N_{cnt}} \boldsymbol{f}_m - m\boldsymbol{g} + \sum_{m=1}^{N_{ex}} \boldsymbol{f}_m^{ex} && (2.17)
\end{aligned}
$$

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

:eom-rot-task-value *&key (update? t)* [method]

$$
\begin{aligned}
\boldsymbol{e}^{eom\text{-}rot}(\boldsymbol{q}) &= \boldsymbol{e}^{eom\text{-}rot}(\boldsymbol{\theta}, \hat{\boldsymbol{w}}, \boldsymbol{\phi}) && (2.18)\\
&= \sum_{m=1}^{N_{cnt}} \left\{ \left(\boldsymbol{p}_m^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}_G(\boldsymbol{\theta}, \boldsymbol{\phi}) \right) \times \boldsymbol{f}_m + \boldsymbol{n}_m \right\} \\
&\quad + \sum_{m=1}^{N_{ex}} \left\{ \left(\boldsymbol{p}_m^{ex}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}_G(\boldsymbol{\theta}, \boldsymbol{\phi}) \right) \times \boldsymbol{f}_m^{ex} + \boldsymbol{n}_m^{ex} \right\} && (2.19)
\end{aligned}
$$

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

:torque-task-value *&key (update? t)* [method]

$$
\begin{aligned}
\boldsymbol{e}^{trq}(\boldsymbol{q}) &= \boldsymbol{e}^{trq}(\boldsymbol{\theta}, \hat{\boldsymbol{w}}, \boldsymbol{\tau}, \boldsymbol{\phi}) && (2.20)\\
&= \boldsymbol{\tau} + \sum_{m=1}^{N_{cnt}} \boldsymbol{\tau}_m^{cnt}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{\tau}^{grav}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \sum_{m=1}^{N_{ex}} \boldsymbol{\tau}_m^{ex}(\boldsymbol{\theta}, \boldsymbol{\phi}) && (2.21)\\
&= \boldsymbol{\tau} + \sum_{m=1}^{N_{cnt}} \boldsymbol{J}_{drive\text{-}joint,m}^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi})^T \boldsymbol{w}_m - \boldsymbol{\tau}^{grav}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \sum_{m=1}^{N_{ex}} \boldsymbol{J}_{drive\text{-}joint,m}^{ex}(\boldsymbol{\theta}, \boldsymbol{\phi})^T \boldsymbol{w}_m^{ex} && (2.22)
\end{aligned}
$$

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

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

:posture-task-value *&key (update? t)* [method]

$$
\begin{aligned}
\boldsymbol{e}^{posture}(\boldsymbol{q}) &= \boldsymbol{e}^{posture}(\boldsymbol{\theta}) && (2.23)\\
&= k_{posture}\left(\bar{\boldsymbol{\theta}}^{trg} - \bar{\boldsymbol{\theta}}\right) && (2.24)
\end{aligned}
$$

$\bar{\boldsymbol{\theta}}^{trg}, \bar{\boldsymbol{\theta}}$ 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 $\boldsymbol{e}^{posture}(\boldsymbol{q}) \in \mathbb{R}^{N_{posture\text{-}joint}}$

:task-value *&key (update? t)* [method]

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

:kinematics-task-jacobian-with-theta [method]

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\theta}} &= \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}_1}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{e}^{kin}_2}{\partial \boldsymbol{\theta}} \\ \vdots \\ \frac{\partial \boldsymbol{e}^{kin}_{N_{kin}}}{\partial \boldsymbol{\theta}} \end{pmatrix} && (2.25)\\
\frac{\partial \boldsymbol{e}^{kin}_m}{\partial \boldsymbol{\theta}} &= K_{kin}\left\{\boldsymbol{J}^{kin\text{-}trg}_{\theta,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{J}^{kin\text{-}att}_{\theta,m}(\boldsymbol{\theta}, \boldsymbol{\phi})\right\} \quad (m = 1, 2, \cdots, N_{kin}) && (2.26)
\end{aligned}
$$

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

:kinematics-task-jacobian-with-phi [method]

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} &= \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}_1}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{kin}_2}{\partial \boldsymbol{\phi}} \\ \vdots \\ \frac{\partial \boldsymbol{e}^{kin}_{N_{kin}}}{\partial \boldsymbol{\phi}} \end{pmatrix} && (2.27)\\
\frac{\partial \boldsymbol{e}^{kin}_m}{\partial \boldsymbol{\phi}} &= K_{kin}\left\{\boldsymbol{J}^{kin\text{-}trg}_{\phi,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{J}^{kin\text{-}att}_{\phi,m}(\boldsymbol{\theta}, \boldsymbol{\phi})\right\} \quad (m = 1, 2, \cdots, N_{kin}) && (2.28)
\end{aligned}
$$

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

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

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \hat{\boldsymbol{w}}} &= \begin{pmatrix} \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{f}_1} & \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{n}_1} & \cdots & \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{f}_{N_{cnt}}} & \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{n}_{N_{cnt}}} \end{pmatrix} && (2.29)\\
&= \begin{pmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 & \cdots & \boldsymbol{I}_3 & \boldsymbol{O}_3 \end{pmatrix} && (2.30)
\end{aligned}
$$

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

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

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\theta}} &= \sum_{m=1}^{N_{cnt}} \left\{ -[\boldsymbol{f}_m \times] \left(\boldsymbol{J}^{cnt\text{-}trg}_{\theta,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \boldsymbol{J}_{G\theta}(\boldsymbol{\theta},\boldsymbol{\phi}) \right) \right\} \\
&\quad + \sum_{m=1}^{N_{ex}} \left\{ -[\boldsymbol{f}^{ex}_m \times] \left(\boldsymbol{J}^{ex}_{\theta,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \boldsymbol{J}_{G\theta}(\boldsymbol{\theta},\boldsymbol{\phi}) \right) \right\} \qquad (2.31) \\
&= \left[\left(\sum_{m=1}^{N_{cnt}} \boldsymbol{f}_m + \sum_{m=1}^{N_{ex}} \boldsymbol{f}^{ex}_m \right) \times \right] \boldsymbol{J}_{G\theta}(\boldsymbol{\theta},\boldsymbol{\phi}) \\
&\quad - \sum_{m=1}^{N_{cnt}} [\boldsymbol{f}_m \times] \boldsymbol{J}^{cnt\text{-}trg}_{\theta,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \sum_{m=1}^{N_{ex}} [\boldsymbol{f}^{ex}_m \times] \boldsymbol{J}^{ex}_{\theta,m}(\boldsymbol{\theta},\boldsymbol{\phi}) \qquad (2.32)
\end{aligned}
$$

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

$$
\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\theta}} = [m\boldsymbol{g} \times] \boldsymbol{J}_{G\theta}(\boldsymbol{\theta},\boldsymbol{\phi}) - \sum_{m=1}^{N_{cnt}} [\boldsymbol{f}_m \times] \boldsymbol{J}^{cnt\text{-}trg}_{\theta,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \sum_{m=1}^{N_{ex}} [\boldsymbol{f}^{ex}_m \times] \boldsymbol{J}^{ex}_{\theta,m}(\boldsymbol{\theta},\boldsymbol{\phi}) \qquad (2.33)
$$

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

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

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \hat{\boldsymbol{w}}} &= \begin{pmatrix} \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{f}_1} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{n}_1} & \cdots & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{f}_{N_{cnt}}} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{n}_{N_{cnt}}} \end{pmatrix} \qquad (2.34) \\
\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{f}_m} &= \left[\left(\boldsymbol{p}^{cnt\text{-}trg}_m(\boldsymbol{\theta},\boldsymbol{\phi}) - \boldsymbol{p}_G(\boldsymbol{\theta},\boldsymbol{\phi}) \right) \times \right] \quad (m = 1, 2, \cdots, N_{cnt}) \qquad (2.35) \\
\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{n}_m} &= \boldsymbol{I}_3 \quad (m = 1, 2, \cdots, N_{cnt}) \qquad (2.36)
\end{aligned}
$$

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

:eom-rot-task-jacobian-with-phi [method]

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} &= \sum_{m=1}^{N_{cnt}} \left\{ -[\boldsymbol{f}_m \times] \left(\boldsymbol{J}^{cnt\text{-}trg}_{\phi,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \boldsymbol{J}_{G\phi}(\boldsymbol{\theta},\boldsymbol{\phi}) \right) \right\} \\
&\quad + \sum_{m=1}^{N_{ex}} \left\{ -[\boldsymbol{f}^{ex}_m \times] \left(\boldsymbol{J}^{ex}_{\phi,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \boldsymbol{J}_{G\phi}(\boldsymbol{\theta},\boldsymbol{\phi}) \right) \right\} \qquad (2.37) \\
&= \left[\left(\sum_{m=1}^{N_{cnt}} \boldsymbol{f}_m + \sum_{m=1}^{N_{ex}} \boldsymbol{f}^{ex}_m \right) \times \right] \boldsymbol{J}_{G\phi}(\boldsymbol{\theta},\boldsymbol{\phi}) \\
&\quad - \sum_{m=1}^{N_{cnt}} [\boldsymbol{f}_m \times] \boldsymbol{J}^{cnt\text{-}trg}_{\phi,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \sum_{m=1}^{N_{ex}} [\boldsymbol{f}^{ex}_m \times] \boldsymbol{J}^{ex}_{\phi,m}(\boldsymbol{\theta},\boldsymbol{\phi}) \qquad (2.38)
\end{aligned}
$$

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

$$
\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} = [m\boldsymbol{g} \times] \boldsymbol{J}_{G\phi}(\boldsymbol{\theta},\boldsymbol{\phi}) - \sum_{m=1}^{N_{cnt}} [\boldsymbol{f}_m \times] \boldsymbol{J}^{cnt\text{-}trg}_{\phi,m}(\boldsymbol{\theta},\boldsymbol{\phi}) - \sum_{m=1}^{N_{ex}} [\boldsymbol{f}^{ex}_m \times] \boldsymbol{J}^{ex}_{\phi,m}(\boldsymbol{\theta},\boldsymbol{\phi}) \qquad (2.39)
$$

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

:torque-task-jacobian-with-theta [method]

$$\frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\theta}} = \sum_{m=1}^{N_{cnt}} \frac{\partial \boldsymbol{\tau}_m^{cnt}}{\partial \boldsymbol{\theta}} - \frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}} + \sum_{m=1}^{N_{ex}} \frac{\partial \boldsymbol{\tau}_m^{ex}}{\partial \boldsymbol{\theta}} \tag{2.40}$$

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

:torque-task-jacobian-with-wrench [method]

$$\frac{\partial \boldsymbol{e}^{trq}}{\partial \hat{\boldsymbol{w}}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{w}_1} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{w}_2} & \cdots & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{w}_{N_{cnt}}} \end{pmatrix} \tag{2.41}$$

$$\frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{w}_m} = \boldsymbol{J}_{drive\text{-}joint,m}^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi})^T \quad (m = 1, 2, \cdots, N_{cnt}) \tag{2.42}$$

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

:torque-task-jacobian-with-phi [method]

$$\frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\phi}} = \sum_{m=1}^{N_{cnt}} \frac{\partial \boldsymbol{\tau}_m^{cnt}}{\partial \boldsymbol{\phi}} - \frac{\partial \boldsymbol{\tau}_m^{grav}}{\partial \boldsymbol{\phi}} + \sum_{m=1}^{N_{ex}} \frac{\partial \boldsymbol{\tau}_m^{ex}}{\partial \boldsymbol{\phi}} \tag{2.43}$$

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

:torque-task-jacobian-with-torque [method]

$$\frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\tau}} = \boldsymbol{I}_{N_{drive\text{-}joint}} \tag{2.44}$$

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

:posture-task-jacobian-with-theta *&key (update? nil)* [method]

$$\left(\frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{\theta}}\right)_{i,j} = \begin{cases} -k_{posture} & (\mathcal{J}_{posture,i} = \mathcal{J}_{var,j}) \\ 0 & \text{otherwise} \end{cases} \tag{2.45}$$

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

:variant-task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{var}} = \begin{array}{c} \\ 6N_{kin} \\ 3 \\ 3 \\ N_{drive\text{-}joint} \\ N_{posture\text{-}joint} \end{array} \begin{array}{c} \begin{array}{ccc} N_{var\text{-}joint} & 6N_{cnt} & N_{drive\text{-}joint} \end{array} \\ \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\theta}} & & \\ & \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \hat{\boldsymbol{w}}} & \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \hat{\boldsymbol{w}}} & \\ \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \hat{\boldsymbol{w}}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\tau}} \\ \frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{\theta}} & & \end{pmatrix} \end{array} \tag{2.46}$$

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

:invariant-task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{invar}} = \begin{array}{cc} & N_{invar\text{-}joint} \\ \begin{array}{l} 6N_{kin} \\ 3 \\ 3 \\ N_{drive\text{-}joint} \\ N_{posture\text{-}joint} \end{array} & \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} \\ \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\phi}} \\ \\ \end{pmatrix} \end{array} \tag{2.47}$$

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

:task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{var}} & \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{invar}} \end{pmatrix} \tag{2.48}$$

$$= \begin{array}{cc} & \begin{array}{cccc} N_{var\text{-}joint} & 6N_{cnt} & N_{drive\text{-}joint} & N_{invar\text{-}joint} \end{array} \\ \begin{array}{l} 6N_{kin} \\ 3 \\ 3 \\ N_{drive\text{-}joint} \\ N_{posture\text{-}joint} \end{array} & \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\theta}} & & & \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} \\ & \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \hat{\boldsymbol{w}}} & & \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \hat{\boldsymbol{w}}} & & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \hat{\boldsymbol{w}}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\tau}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{\theta}} & & & \end{pmatrix} \end{array} \tag{2.49}$$

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

:theta-max-vector *&key (update? nil)* [method]

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

:theta-min-vector *&key (update? nil)* [method]

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

:delta-theta-limit-vector *&key (update? nil)* [method]

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

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

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

$$\begin{cases} \boldsymbol{\theta}_{min} \leq \boldsymbol{\theta} + \Delta\boldsymbol{\theta} \leq \boldsymbol{\theta}_{max} \\ -\Delta\boldsymbol{\theta}_{limit} \leq \Delta\boldsymbol{\theta} \leq \Delta\boldsymbol{\theta}_{limit} \quad (\text{if} \quad \Delta\boldsymbol{\theta}_{limit} \quad \text{is set}) \end{cases} \tag{2.50}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \\ \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{\theta} \geq \begin{pmatrix} \boldsymbol{\theta}_{min} - \boldsymbol{\theta} \\ -(\boldsymbol{\theta}_{max} - \boldsymbol{\theta}) \\ -\Delta\boldsymbol{\theta}_{limit} \\ -\Delta\boldsymbol{\theta}_{limit} \end{pmatrix} \tag{2.51}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{\theta}} \Delta\boldsymbol{\theta} \geq \boldsymbol{d}_{\boldsymbol{\theta}} \tag{2.52}$$

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

:theta-inequality-constraint-vector *&key (update? t)* [method]

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

:wrench-inequality-constraint-matrix *&key (update? t)* [method]

接触レンチ $\boldsymbol{w} \in \mathbb{R}^6$ 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．

$$\boldsymbol{C}_w \boldsymbol{w} \geq \boldsymbol{d}_w \tag{2.53}$$

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

$$\boldsymbol{C}_{w,m}(\boldsymbol{w}_m + \Delta\boldsymbol{w}_m) \geq \boldsymbol{d}_{w,m} \quad (m = 1, 2, \cdots, N_{cnt}) \tag{2.54}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{w,m}\Delta\boldsymbol{w}_m \geq \boldsymbol{d}_{w,m} - \boldsymbol{C}_{w,m}\boldsymbol{w}_m \quad (m = 1, 2, \cdots, N_{cnt}) \tag{2.55}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C}_{w,1} & & & \\ & \boldsymbol{C}_{w,2} & & \\ & & \ddots & \\ & & & \boldsymbol{C}_{w,N_{cnt}} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{w}_1 \\ \Delta\boldsymbol{w}_2 \\ \vdots \\ \Delta\boldsymbol{w}_{N_{cnt}} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_{w,1} - \boldsymbol{C}_{w,1}\boldsymbol{w}_1 \\ \boldsymbol{d}_{w,2} - \boldsymbol{C}_{w,2}\boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{d}_{w,N_{cnt}} - \boldsymbol{C}_{w,N_{cnt}}\boldsymbol{w}_{N_{cnt}} \end{pmatrix} \tag{2.56}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\hat{w}}\Delta\hat{\boldsymbol{w}} \geq \boldsymbol{d}_{\hat{w}} \tag{2.57}$$

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

:wrench-inequality-constraint-vector *&key (update? t)* [method]

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

:torque-max-vector *&key (update? nil)* [method]

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

:torque-min-vector *&key (update? nil)* [method]

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

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

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau} + \Delta\boldsymbol{\tau} \leq \boldsymbol{\tau}_{max} \tag{2.58}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{\tau} \geq \begin{pmatrix} \boldsymbol{\tau}_{min} - \boldsymbol{\tau} \\ -(\boldsymbol{\tau}_{max} - \boldsymbol{\tau}) \end{pmatrix} \tag{2.59}$$

$$\Leftrightarrow \quad \boldsymbol{C_\tau}\Delta\boldsymbol{\tau} \geq \boldsymbol{d_\tau} \tag{2.60}$$

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

:torque-inequality-constraint-vector *&key (update? t)* [method] return $\boldsymbol{d_\tau} := \begin{pmatrix} \boldsymbol{\tau}_{min} - \boldsymbol{\tau} \\ -(\boldsymbol{\tau}_{max} - \boldsymbol{\tau}) \end{pmatrix} \in \mathbb{R}^{2N_{drive\text{-}joint}}$

:phi-max-vector *&key (update? nil)* [method]

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

:phi-min-vector *&key (update? nil)* [method]

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

:delta-phi-limit-vector *&key (update? nil)* [method]

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

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

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

$$\begin{cases} \boldsymbol{\phi}_{min} \leq \boldsymbol{\phi} + \Delta\boldsymbol{\phi} \leq \boldsymbol{\phi}_{max} \\ -\Delta\boldsymbol{\phi}_{limit} \leq \Delta\boldsymbol{\phi} \leq \Delta\boldsymbol{\phi}_{limit} \quad (\text{if} \ \ \Delta\boldsymbol{\phi}_{limit} \ \ \text{is set}) \end{cases} \tag{2.61}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \\ \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{\phi} \geq \begin{pmatrix} \boldsymbol{\phi}_{min} - \boldsymbol{\phi} \\ -(\boldsymbol{\phi}_{max} - \boldsymbol{\phi}) \\ -\Delta\boldsymbol{\phi}_{limit} \\ -\Delta\boldsymbol{\phi}_{limit} \end{pmatrix} \tag{2.62}$$

$$\Leftrightarrow \quad \boldsymbol{C_\phi}\Delta\boldsymbol{\phi} \geq \boldsymbol{d_\phi} \tag{2.63}$$

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

:phi-inequality-constraint-vector *&key (update? t)* [method]

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

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

$$\begin{cases} \boldsymbol{C}_\theta \Delta\boldsymbol{\theta} \geq \boldsymbol{d}_\theta \\ \boldsymbol{C}_{\hat{w}} \Delta\hat{\boldsymbol{w}} \geq \boldsymbol{d}_{\hat{w}} \\ \boldsymbol{C}_\tau \Delta\boldsymbol{\tau} \geq \boldsymbol{d}_\tau \end{cases} \tag{2.64}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C}_\theta & & \\ & \boldsymbol{C}_{\hat{w}} & \\ & & \boldsymbol{C}_\tau \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\theta} \\ \Delta\hat{\boldsymbol{w}} \\ \Delta\boldsymbol{\tau} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_\theta \\ \boldsymbol{d}_{\hat{w}} \\ \boldsymbol{d}_\tau \end{pmatrix} \tag{2.65}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{var}\Delta\boldsymbol{q}_{var} \geq \boldsymbol{d}_{var} \tag{2.66}$$

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

:variant-config-inequality-constraint-vector *&key (update? t)* [method] return $\boldsymbol{d}_{var} := \begin{pmatrix} \boldsymbol{d}_{\theta} \\ \boldsymbol{d}_{\hat{w}} \\ \boldsymbol{d}_{\tau} \end{pmatrix} \in \mathbb{R}^{N_{var\text{-}ineq}}$

:invariant-config-inequality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{C}_{\phi} \Delta \boldsymbol{\phi} \geq \boldsymbol{d}_{\phi} \tag{2.67}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{invar} \Delta \boldsymbol{q}_{invar} \geq \boldsymbol{d}_{invar} \tag{2.68}$$

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

:invariant-config-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{invar} := \boldsymbol{d}_{\phi} \in \mathbb{R}^{N_{invar\text{-}ineq}}$

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

$$\begin{cases} \boldsymbol{C}_{var} \Delta \boldsymbol{q}_{var} \geq \boldsymbol{d}_{var} \\ \boldsymbol{C}_{invar} \Delta \boldsymbol{q}_{invar} \geq \boldsymbol{d}_{invar} \\ \boldsymbol{C}_{col} \begin{pmatrix} \Delta \boldsymbol{q}_{var} \\ \Delta \boldsymbol{q}_{invar} \end{pmatrix} \geq \boldsymbol{d}_{col} \end{cases} \tag{2.69}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C}_{var} & \\ & \boldsymbol{C}_{invar} \\ \hdashline \multicolumn{2}{c}{\boldsymbol{C}_{col}} \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{q}_{var} \\ \Delta \boldsymbol{q}_{invar} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_{var} \\ \boldsymbol{d}_{invar} \\ \boldsymbol{d}_{col} \end{pmatrix} \tag{2.70}$$

$$\Leftrightarrow \quad \boldsymbol{C} \Delta \boldsymbol{q} \geq \boldsymbol{d} \tag{2.71}$$

return $\boldsymbol{C} := \begin{pmatrix} \boldsymbol{C}_{var} & \\ & \boldsymbol{C}_{invar} \\ \hdashline \multicolumn{2}{c}{\boldsymbol{C}_{col}} \end{pmatrix} \in \mathbb{R}^{N_{ineq} \times dim(\boldsymbol{q})}$

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

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

:variant-config-equality-constraint-matrix *&key (update? nil)* [method]

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

:variant-config-equality-constraint-vector *&key (update? t)* [method]

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

:invariant-config-equality-constraint-matrix *&key (update? nil)* [method]

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

:invariant-config-equality-constraint-vector *&key (update? t)* [method]

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

:config-equality-constraint-matrix *&key (update? nil)* [method] return $\boldsymbol{A} \in \mathbb{R}^{0 \times dim(\boldsymbol{q})}$ (no equality constraint)

:config-equality-constraint-vector *&key (update? t)* [method]

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

:torque-regular-matrix *&key (update? nil)*
(only-variant? nil) [method]

二次形式の正則化項として次式を考える．

$$F_{tau}(\boldsymbol{q}) = \left\| \frac{\boldsymbol{\tau}}{\boldsymbol{\tau}_{max}} \right\|^2 \text{ (ベクトルの要素ごとの割り算を表す)} \tag{2.72}$$

$$= \boldsymbol{\tau}^T \bar{\boldsymbol{W}}_{trq} \boldsymbol{\tau} \tag{2.73}$$

ここで,

$$\bar{\boldsymbol{W}}_{trq} := \begin{pmatrix} \frac{1}{\tau_{max,1}^2} & & & \\ & \frac{1}{\tau_{max,2}^2} & & \\ & & \ddots & \\ & & & \frac{1}{\tau_{max,N_{drive\text{-}joint}}^2} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{\tau}) \times dim(\boldsymbol{\tau})} \tag{2.74}$$

only-variant? is true:

$$\boldsymbol{W}_{trq} := \begin{array}{c c} & \begin{array}{ccc} dim(\boldsymbol{\theta}) & dim(\hat{\boldsymbol{w}}) & dim(\boldsymbol{\tau}) \end{array} \\ \begin{array}{c} dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\boldsymbol{\tau}) \end{array} & \begin{pmatrix} & & \\ & & \\ & & \bar{\boldsymbol{W}}_{trq} \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q}_{var}) \times dim(\boldsymbol{q}_{var})} \tag{2.75}$$

otherwise:

$$\boldsymbol{W}_{trq} := \begin{array}{c c} & \begin{array}{cccc} dim(\boldsymbol{\theta}) & dim(\hat{\boldsymbol{w}}) & dim(\boldsymbol{\tau}) & dim(\boldsymbol{\phi}) \end{array} \\ \begin{array}{c} dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\boldsymbol{\tau}) \\ dim(\boldsymbol{\phi}) \end{array} & \begin{pmatrix} & & & \\ & & & \\ & & \bar{\boldsymbol{W}}_{trq} & \\ & & & \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q}) \times dim(\boldsymbol{q})} \tag{2.76}$$

return $\boldsymbol{W}_{trq}$

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

$$\bar{\boldsymbol{v}}_{trq} := \bar{\boldsymbol{W}}_{trq} \boldsymbol{\tau} \tag{2.77}$$

$$= \begin{pmatrix} \frac{\tau_1}{\tau_{max,1}^2} \\ \frac{\tau_2}{\tau_{max,2}^2} \\ \vdots \\ \frac{\tau_{dim(\boldsymbol{\tau})}}{\tau_{max,dim(\boldsymbol{\tau})}^2} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{\tau})} \tag{2.78}$$

only-variant? is true:

$$\boldsymbol{v}_{trq} := \begin{array}{c c} & 1 \\ \begin{array}{c} dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\boldsymbol{\tau}) \end{array} & \begin{pmatrix} \\ \\ \bar{\boldsymbol{v}}_{trq} \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q}_{var})} \tag{2.79}$$

otherwise:

$$\boldsymbol{v}_{trq} := \begin{array}{r} \\ dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\boldsymbol{\tau}) \\ dim(\boldsymbol{\phi}) \end{array} \begin{array}{c} 1 \\ \begin{pmatrix} \\ \\ \bar{\boldsymbol{v}}_{trq} \\ \\ \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q})} \tag{2.80}$$

return $\boldsymbol{v}_{trq}$

:torque-ratio [method]

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

:wrench-maximize-regular-vector *&key (update? nil)*
(only-variant? nil) [method]

$$\bar{\boldsymbol{v}}_{w\text{-}max} \quad := \quad \begin{pmatrix} \boldsymbol{d}_1^{w\text{-}max} \\ \boldsymbol{d}_2^{w\text{-}max} \\ \vdots \\ \boldsymbol{d}_{N_{cnt}}^{w\text{-}max} \end{pmatrix} \in \mathbb{R}^{dim(\hat{\boldsymbol{w}})} \tag{2.81}$$

only-variant? is true:

$$\boldsymbol{v}_{w\text{-}max} := \begin{array}{r} \\ dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\boldsymbol{\tau}) \end{array} \begin{array}{c} 1 \\ \begin{pmatrix} \\ \bar{\boldsymbol{v}}_{w\text{-}max} \\ \\ \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q}_{var})} \tag{2.82}$$

otherwise:

$$\boldsymbol{v}_{w\text{-}max} := \begin{array}{r} \\ dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\boldsymbol{\tau}) \\ dim(\boldsymbol{\phi}) \end{array} \begin{array}{c} 1 \\ \begin{pmatrix} \\ \bar{\boldsymbol{v}}_{w\text{-}max} \\ \\ \\ \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q})} \tag{2.83}$$

return $\boldsymbol{v}_{w\text{-}max}$

:regular-matrix [method]

$$\boldsymbol{W}_{reg} := \min(k_{max}, k_{coeff}\|\boldsymbol{e}\|^2 + k_{off})\boldsymbol{I} + k_{trq}\boldsymbol{W}_{trq} \tag{2.84}$$

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

:regular-vector [method]

$$\boldsymbol{v}_{reg} := k_{trq}\boldsymbol{v}_{trq} + k_{w\text{-}max}\boldsymbol{v}_{w\text{-}max} \tag{2.85}$$

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

:update-collision-inequality-constraint [method]

リンク 1 とリンク 2 の最近点を $\boldsymbol{p}_1, \boldsymbol{p}_2$ とする．リンク 1 とリンク 2 が干渉しない条件を，最近点 $\boldsymbol{p}_1, \boldsymbol{p}_2$ 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．

$$\boldsymbol{d}_{12}^T(\boldsymbol{p}_1 - \boldsymbol{p}_2) \geq d_{margin} \tag{2.86}$$
$$\text{where} \quad \boldsymbol{d}_{12} = \boldsymbol{p}_1 - \boldsymbol{p}_2 \tag{2.87}$$

コンフィギュレーションが $\Delta\boldsymbol{q}$ 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．

$$\boldsymbol{d}_{12}^T\left\{(\boldsymbol{p}_1 + \Delta\boldsymbol{p}_1) - (\boldsymbol{p}_2 + \Delta\boldsymbol{p}_2)\right\} \geq d_{margin} \tag{2.88}$$
$$\text{where} \quad \Delta\boldsymbol{p}_1 = \boldsymbol{J}_{\theta,1}\Delta\boldsymbol{\theta} + \boldsymbol{J}_{\phi,1}\Delta\boldsymbol{\phi} \tag{2.89}$$
$$\Delta\boldsymbol{p}_2 = \boldsymbol{J}_{\theta,2}\Delta\boldsymbol{\theta} + \boldsymbol{J}_{\phi,2}\Delta\boldsymbol{\phi} \tag{2.90}$$
$$\boldsymbol{J}_{\theta,i} = \frac{\partial\boldsymbol{p}_i}{\partial\boldsymbol{\theta}}, \quad \boldsymbol{J}_{\phi,i} = \frac{\partial\boldsymbol{p}_i}{\partial\boldsymbol{\phi}} \quad (i = 1, 2) \tag{2.91}$$

これは以下のように変形される．

$$\boldsymbol{d}_{12}^T\left\{(\boldsymbol{p}_1 + \boldsymbol{J}_{\theta,1}\Delta\boldsymbol{\theta} + \boldsymbol{J}_{\phi,1}\Delta\boldsymbol{\phi}) - (\boldsymbol{p}_2 + \boldsymbol{J}_{\theta,2}\Delta\boldsymbol{\theta} + \boldsymbol{J}_{\phi,2}\Delta\boldsymbol{\phi})\right\} \geq d_{margin} \tag{2.92}$$
$$\Leftrightarrow \quad \boldsymbol{d}_{12}^T(\boldsymbol{J}_{\theta,1} - \boldsymbol{J}_{\theta,2})\Delta\boldsymbol{\theta} + \boldsymbol{d}_{12}^T(\boldsymbol{J}_{\phi,1} - \boldsymbol{J}_{\phi,2})\Delta\boldsymbol{\phi} \geq -(\boldsymbol{d}_{12}^T(\boldsymbol{p}_1 - \boldsymbol{p}_2) - d_{margin}) \tag{2.93}$$
$$\Leftrightarrow \quad \boldsymbol{c}_{col,var}^T\Delta\boldsymbol{\theta} + \boldsymbol{c}_{col,invar}^T\Delta\boldsymbol{\phi} \geq d_{col} \tag{2.94}$$
$$\text{where} \quad \boldsymbol{c}_{col,var}^T = \boldsymbol{d}_{12}^T(\boldsymbol{J}_{\theta,1} - \boldsymbol{J}_{\theta,2}) \tag{2.95}$$
$$\boldsymbol{c}_{col,invar}^T = \boldsymbol{d}_{12}^T(\boldsymbol{J}_{\phi,1} - \boldsymbol{J}_{\phi,2}) \tag{2.96}$$
$$d_{col} = -(\boldsymbol{d}_{12}^T(\boldsymbol{p}_1 - \boldsymbol{p}_2) - d_{margin}) \tag{2.97}$$

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 $\boldsymbol{c}_{col,var,i}^T, \boldsymbol{c}_{col,invar,i}^T, d_{col,i}$ とする．$i = 1, 2, \cdots, N_{col}$ 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．

$$\begin{pmatrix} \boldsymbol{C}_{col,\theta} & \boldsymbol{C}_{col,\phi} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\theta} \\ \Delta\boldsymbol{\phi} \end{pmatrix} \geq \boldsymbol{d}_{col} \tag{2.98}$$
$$\boldsymbol{C}_{col,\theta} := \begin{pmatrix} \boldsymbol{c}_{col,var,1}^T \\ \vdots \\ \boldsymbol{c}_{col,var,N_{col}}^T \end{pmatrix} \in \mathbb{R}^{N_{col} \times dim(\boldsymbol{\theta})} \tag{2.99}$$
$$\boldsymbol{C}_{col,\phi} := \begin{pmatrix} \boldsymbol{c}_{col,invar,1}^T \\ \vdots \\ \boldsymbol{c}_{col,invar,N_{col}}^T \end{pmatrix} \in \mathbb{R}^{N_{col} \times dim(\boldsymbol{\phi})}, \tag{2.100}$$
$$\boldsymbol{d}_{col} := \begin{pmatrix} d_{col,1} \\ \vdots \\ d_{col,N_{col}} \end{pmatrix} \in \mathbb{R}^{N_{col}} \tag{2.101}$$

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

:collision-theta-inequality-constraint-matrix [method] return $\boldsymbol{C}_{col,\theta} \in \mathbb{R}^{N_{col} \times dim(\boldsymbol{\theta})}$

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

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

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

$$\boldsymbol{C}_{col} := N_{col} \begin{array}{c} \begin{array}{cccc} dim(\boldsymbol{\theta}) & dim(\hat{\boldsymbol{w}}) & dim(\boldsymbol{\tau}) & dim(\boldsymbol{\phi}) \end{array} \\ \begin{pmatrix} \boldsymbol{C}_{col,\theta} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{C}_{col,\phi} \end{pmatrix} \end{array} \tag{2.102}$$

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

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

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

: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 $dim(\boldsymbol{q}_{var})$)
(_dim-invariant-config $dim(\boldsymbol{q}_{invar})$)
(_dim-config $dim(\boldsymbol{q})$)
(_dim-task $dim(\boldsymbol{e})$)
(_norm-regular-scale-max k_{max})
(_norm-regular-scale-offset k_{off})
(_adjacent-regular-scale-list $k_{adj}^{(1)}, k_{adj}^{(2)}, \cdots, k_{adj}^{(L-1)}$)
(_torque-regular-scale k_{trq})
(_task-jacobi buffer for $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}}$)

軌道コンフィギュレーション $\boldsymbol{q}$ と軌道タスク関数 $\boldsymbol{e}(\boldsymbol{q})$ 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 $\boldsymbol{q}$ の取得・更新，タスク関数 $\boldsymbol{e}(\boldsymbol{q})$ の取得，タスク関数のヤコビ行列 $\frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ の取得，コンフィギュレーションの等式・不等式制約 $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{C}, \boldsymbol{d}$ 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-list $k_{adj}^{(l)}$ 隣接コンフィギュレーション正則化の重みのリスト

  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 $\boldsymbol{q}$ は以下から構成される．

$$\boldsymbol{q} := \begin{pmatrix} \boldsymbol{q}_{var}^{(1)T} & \boldsymbol{q}_{var}^{(2)T} & \cdots & \boldsymbol{q}_{var}^{(L)T} & \boldsymbol{q}_{invar}^{T} \end{pmatrix}^T \tag{2.103}$$

ここで,

$$\boldsymbol{q}_{invar} := \boldsymbol{q}_{invar}^{(1)} = \boldsymbol{q}_{invar}^{(2)} = \cdots = \boldsymbol{q}_{invar}^{(L)} \tag{2.104}$$

$\boldsymbol{q}_{var}^{(l)}, \boldsymbol{q}_{invar}^{(l)}$ $(l = 1, 2, \cdots, L)$ 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 $\boldsymbol{e}(\boldsymbol{q})$ は以下から構成される．

$$\boldsymbol{e}(\boldsymbol{q}) := \begin{pmatrix} \boldsymbol{e}^{(1)T}(\boldsymbol{q}_{var}^{(1)}, \boldsymbol{q}_{invar}) & \boldsymbol{e}^{(2)T}(\boldsymbol{q}_{var}^{(2)}, \boldsymbol{q}_{invar}) & \cdots & \boldsymbol{e}^{(L)T}(\boldsymbol{q}_{var}^{(L)}, \boldsymbol{q}_{invar}) \end{pmatrix}^T \tag{2.105}$$

$\boldsymbol{e}^{(l)}(\boldsymbol{q}_{var}^{(l)}, \boldsymbol{q}_{invar})$ $(l = 1, 2, \cdots, L)$ 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)
(adjacent-regular-scale-list)
(torque-regular-scale 0.001)

Initialize instance

:instant-config-task-list [method]

return instant-config-task-list

:dim-variant-config [method]

return $dim(\boldsymbol{q_{var}}) := \sum_{l=1}^{L} dim(\boldsymbol{q}_{var}^{(l)})$

:dim-invariant-config [method]

return $dim(\boldsymbol{q_{invar}}) := dim(\boldsymbol{q}_{invar}^{(l)})$ ($l = 1, 2, \cdots, L$ で同じ)

:dim-config [method]

return $dim(\boldsymbol{q}) := dim(\boldsymbol{q_{var}}) + dim(\boldsymbol{q_{invar}})$

:dim-task [method]

return $dim(\boldsymbol{e}) := \sum_{l=1}^{L} dim(\boldsymbol{e}^{(l)})$

:variant-config-vector [method]

return $\boldsymbol{q_{var}} := \begin{pmatrix} \boldsymbol{q}_{var}^{(1)} \\ \boldsymbol{q}_{var}^{(2)} \\ \vdots \\ \boldsymbol{q}_{var}^{(L)} \end{pmatrix}$

:invariant-config-vector [method]

return $\boldsymbol{q_{invar}} := \boldsymbol{q}_{invar}^{(l)}$ ($l = 1, 2, \cdots, L$ で同じ)

:config-vector [method]

$$\text{return } \boldsymbol{q} := \begin{pmatrix} \boldsymbol{q_{var}} \\ \boldsymbol{q_{invar}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}_{var}^{(1)} \\ \boldsymbol{q}_{var}^{(2)} \\ \vdots \\ \boldsymbol{q}_{var}^{(L)} \\ \boldsymbol{q}_{invar} \end{pmatrix}$$

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

Set $\boldsymbol{q_{var}}$.

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

Set $\boldsymbol{q_{invar}}$.

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

Set $\boldsymbol{q}$.

:task-value *&key (update? t)* [method]

$$\text{return } \boldsymbol{e}(\boldsymbol{q}) := \begin{pmatrix} \boldsymbol{e}^{(1)}(\boldsymbol{q}_{var}^{(1)}, \boldsymbol{q}_{invar}) \\ \boldsymbol{e}^{(2)}(\boldsymbol{q}_{var}^{(2)}, \boldsymbol{q}_{invar}) \\ \vdots \\ \boldsymbol{e}^{(L)}(\boldsymbol{q}_{var}^{(L)}, \boldsymbol{q}_{invar}) \end{pmatrix}$$

:variant-task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{var}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{(1)}}{\partial \boldsymbol{q}_{var}^{(1)}} & & & \boldsymbol{O} \\ & \frac{\partial \boldsymbol{e}^{(2)}}{\partial \boldsymbol{q}_{var}^{(2)}} & & \\ & & \ddots & \\ \boldsymbol{O} & & & \frac{\partial \boldsymbol{e}^{(L)}}{\partial \boldsymbol{q}_{var}^{(L)}} \end{pmatrix} \tag{2.106}$$

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

:invariant-task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{invar}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{(1)}}{\partial \boldsymbol{q}_{invar}} \\ \frac{\partial \boldsymbol{e}^{(2)}}{\partial \boldsymbol{q}_{invar}} \\ \vdots \\ \frac{\partial \boldsymbol{e}^{(L)}}{\partial \boldsymbol{q}_{invar}} \end{pmatrix} \tag{2.107}$$

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

:task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{var}} & \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{invar}} \end{pmatrix} \tag{2.108}$$

$$= \begin{pmatrix} \frac{\partial \boldsymbol{e}^{(1)}}{\partial \boldsymbol{q}_{var}^{(1)}} & & & \boldsymbol{O} & \frac{\partial \boldsymbol{e}^{(1)}}{\partial \boldsymbol{q}_{invar}} \\ & \frac{\partial \boldsymbol{e}^{(2)}}{\partial \boldsymbol{q}_{var}^{(2)}} & & & \frac{\partial \boldsymbol{e}^{(2)}}{\partial \boldsymbol{q}_{invar}} \\ & & \ddots & & \\ \boldsymbol{O} & & & \frac{\partial \boldsymbol{e}^{(L)}}{\partial \boldsymbol{q}_{var}^{(L)}} & \frac{\partial \boldsymbol{e}^{(L)}}{\partial \boldsymbol{q}_{invar}} \end{pmatrix} \tag{2.109}$$

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

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

$$\boldsymbol{C}_{var} := \begin{pmatrix} \boldsymbol{C}_{var}^{(1)} & & & \boldsymbol{O} \\ & \boldsymbol{C}_{var}^{(2)} & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{C}_{var}^{(L)} \end{pmatrix} \tag{2.110}$$

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

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

$$\boldsymbol{d}_{var} := \begin{pmatrix} \boldsymbol{d}_{var}^{(1)} \\ \boldsymbol{d}_{var}^{(2)} \\ \vdots \\ \boldsymbol{d}_{var}^{(L)} \end{pmatrix} \tag{2.111}$$

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

:invariant-config-inequality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{C}_{invar} := \boldsymbol{C}_{invar}^{(l)} \ (l = 1, 2, \cdots, L \text{ で同じ}) \tag{2.112}$$

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

:invariant-config-inequality-constraint-vector *&key (update? t)* [method]

$$\boldsymbol{d}_{invar} := \boldsymbol{d}_{invar}^{(l)} \ (l = 1, 2, \cdots, L \text{ で同じ}) \tag{2.113}$$

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

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

$$\boldsymbol{C} := \begin{pmatrix} \boldsymbol{C}_{var} & \\ & \boldsymbol{C}_{invar} \\ \hdashline \multicolumn{2}{c}{\boldsymbol{C}_{col}} \end{pmatrix} \in \mathbb{R}^{N_{ineq} \times dim(\boldsymbol{q})} \tag{2.114}$$

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

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

$$\boldsymbol{d} := \begin{pmatrix} \boldsymbol{d}_{var} \\ \boldsymbol{d}_{invar} \\ \boldsymbol{d}_{col} \end{pmatrix} \tag{2.115}$$

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

:variant-config-equality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{A}_{var} := \begin{pmatrix} \boldsymbol{A}_{var}^{(1)} & & & \boldsymbol{O} \\ & \boldsymbol{A}_{var}^{(2)} & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{A}_{var}^{(L)} \end{pmatrix} \tag{2.116}$$

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

:variant-config-equality-constraint-vector *&key (update? t)* [method]

$$\boldsymbol{b}_{var} := \begin{pmatrix} \boldsymbol{b}_{var}^{(1)} \\ \boldsymbol{b}_{var}^{(2)} \\ \vdots \\ \boldsymbol{b}_{var}^{(L)} \end{pmatrix} \tag{2.117}$$

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

:invariant-config-equality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{A}_{invar} := \boldsymbol{A}_{invar}^{(l)} \ (l = 1, 2, \cdots, L \text{ で同じ}) \tag{2.118}$$

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

:invariant-config-equality-constraint-vector *&key (update? t)* [method]

$$\boldsymbol{b}_{invar} := \boldsymbol{b}_{invar}^{(l)} \ (l = 1, 2, \cdots, L \text{ で同じ}) \tag{2.119}$$

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

:config-equality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{A} := \begin{pmatrix} \boldsymbol{A}_{var} & \\ & \boldsymbol{A}_{invar} \end{pmatrix} \in \mathbb{R}^{N_{eq} \times dim(\boldsymbol{q})} \tag{2.120}$$

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

:config-equality-constraint-vector *&key (update? t)* [method]

$$\boldsymbol{b} := \begin{pmatrix} \boldsymbol{b}_{var} \\ \boldsymbol{b}_{invar} \end{pmatrix} \tag{2.121}$$

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

:update-collision-inequality-constraint [method]

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

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

$$\begin{array}{rcl} & & \begin{array}{ccc} dim(\boldsymbol{\theta}^{(l)}) & dim(\hat{\boldsymbol{w}}^{(l)}) & dim(\boldsymbol{\tau}^{(l)}) \end{array} \\ \hat{\boldsymbol{C}}^{(l)}_{col,\theta} & := & N^{(l)}_{col} \begin{pmatrix} \boldsymbol{C}^{(l)}_{col,\theta} & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \end{array} \tag{2.122}$$

$$\boldsymbol{C}_{col} := \begin{pmatrix} \hat{\boldsymbol{C}}^{(1)}_{col,\theta} & & & & \boldsymbol{C}^{(1)}_{col,\phi} \\ & \hat{\boldsymbol{C}}^{(2)}_{col,\theta} & & & \boldsymbol{C}^{(2)}_{col,\phi} \\ & & \ddots & & \vdots \\ & & & \hat{\boldsymbol{C}}^{(L)}_{col,\theta} & \boldsymbol{C}^{(L)}_{col,\phi} \end{pmatrix} \tag{2.123}$$

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

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

$$\boldsymbol{d}_{col} := \begin{pmatrix} \boldsymbol{d}^{(1)}_{col} \\ \boldsymbol{d}^{(2)}_{col} \\ \vdots \\ \boldsymbol{d}^{(L)}_{col} \end{pmatrix} \tag{2.124}$$

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

:adjacent-regular-matrix *&key (update? nil)* [method]

二次形式の正則化項として次式を考える.

$$F_{adj}(\boldsymbol{q}) = \sum_{l=1}^{L-1} k^{(l)}_{adj} \|\boldsymbol{\theta}_{l+1} - \boldsymbol{\theta}_l\|^2 \tag{2.125}$$

$$= \boldsymbol{q}^T \boldsymbol{W}_{adj} \boldsymbol{q} \tag{2.126}$$

ここで,

$$\begin{array}{rcl} & & \begin{array}{ccc} dim(\boldsymbol{\theta}^{(l)}) & dim(\hat{\boldsymbol{w}}^{(l)}) & dim(\boldsymbol{\tau}^{(l)}) \end{array} \\ \bar{\boldsymbol{I}}^{(l)}_{adj} & := & \begin{array}{c} dim(\boldsymbol{\theta}^{(l)}) \\ dim(\hat{\boldsymbol{w}}^{(l)}) \\ dim(\boldsymbol{\tau}^{(l)}) \end{array} \begin{pmatrix} k^{(l)}_{adj}\boldsymbol{I} & & \\ & & \\ & & \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{q}^{(l)}_{var}) \times dim(\boldsymbol{q}^{(l)}_{var})} \end{array} \tag{2.127}$$

$$\bar{\boldsymbol{W}}_{adj} := \begin{pmatrix} \bar{\boldsymbol{I}}^{(1)}_{adj} & -\bar{\boldsymbol{I}}^{(1)}_{adj} & & & \boldsymbol{O} \\ -\bar{\boldsymbol{I}}^{(1)}_{adj} & \bar{\boldsymbol{I}}^{(1)}_{adj} + \bar{\boldsymbol{I}}^{(2)}_{adj} & -\bar{\boldsymbol{I}}^{(2)}_{adj} & & \\ & & \ddots & & \\ & & & \bar{\boldsymbol{I}}^{(L-2)}_{adj} + \bar{\boldsymbol{I}}^{(L-1)}_{adj} & -\bar{\boldsymbol{I}}^{(L-1)}_{adj} \\ \boldsymbol{O} & & & -\bar{\boldsymbol{I}}^{(L-1)}_{adj} & \bar{\boldsymbol{I}}^{(L-1)}_{adj} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{q}_{var}) \times dim(\boldsymbol{q}_{var})} \tag{2.128}$$

$$\boldsymbol{W}_{adj} := \begin{pmatrix} \bar{\boldsymbol{W}}_{adj} & \\ & \boldsymbol{O} \end{pmatrix} \tag{2.129}$$

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

:adjacent-regular-vector *&key (update? t)* [method]

$$\boldsymbol{v}_{adj} \quad := \quad \boldsymbol{W}_{adj}\boldsymbol{q} \tag{2.130}$$

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

:torque-regular-matrix *&key (update? nil)* [method]

$$\bar{\boldsymbol{W}}_{trq} \quad := \quad \begin{pmatrix} \boldsymbol{W}_{trq}^{(1)} & & & \boldsymbol{O} \\ & \boldsymbol{W}_{trq}^{(2)} & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{W}_{trq}^{(L)} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{q}_{var}) \times dim(\boldsymbol{q}_{var})} \tag{2.131}$$

$$\boldsymbol{W}_{trq} \quad := \quad \begin{pmatrix} \bar{\boldsymbol{W}}_{trq} & \\ & \boldsymbol{O} \end{pmatrix} \tag{2.132}$$

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

:torque-regular-vector *&key (update? t)* [method]

$$\bar{\boldsymbol{v}}_{trq} \quad := \quad \begin{pmatrix} \boldsymbol{v}_{trq}^{(1)} \\ \boldsymbol{v}_{trq}^{(2)} \\ \vdots \\ \boldsymbol{v}_{trq}^{(L)} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{q}_{var})} \tag{2.133}$$

$$\boldsymbol{v}_{trq} \quad := \quad \begin{pmatrix} \bar{\boldsymbol{v}}_{trq} \\ \boldsymbol{0} \end{pmatrix} \tag{2.134}$$

return $\boldsymbol{v}_{trq} \in \mathbb{R}^{dim(\boldsymbol{q})}$

:regular-matrix [method]

$$\boldsymbol{W}_{reg} := \min(k_{max}, \|\boldsymbol{e}\|^2 + k_{off})\boldsymbol{I} + \boldsymbol{W}_{adj} + k_{trq}\boldsymbol{W}_{trq} \tag{2.135}$$

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

:regular-vector [method]

$$\boldsymbol{v}_{reg} := \boldsymbol{v}_{adj} + k_{trq}\boldsymbol{v}_{trq} \tag{2.136}$$

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

:update-viewer [method]

Update viewer.

:print-status [method]

Print status.

:play-animation *&key* *(robot-env)* [method]
(loop? t)
(visualize-callback-func)
(visualize-force? nil)
(force-color #f(0.8 0.2 0.2))

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 勾配を用いた制約付き非線形最適化

3.1 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 $\boldsymbol{e}(\boldsymbol{q})$)
(_task-jacobian buffer for task jacobian $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}}$)
(_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 supress visualization)
(_no-print? whether to supress 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 $\|\boldsymbol{e}(\boldsymbol{q})\|^2$ を最小にするコンフィギュレーション $\boldsymbol{q}$ を反復計算により求める．

:init *&key* *(config-task)* [method]
(convergence-check-func)

(failure-callback-func)
(pre-process-func)
(post-process-func)
(no-visualize?)
(no-print?)
&allow-other-keys

Initialize instance

:config-task [method]

Return configuration-task instance

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

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:

$$\min_{\Delta \boldsymbol{q}^{(k)}} \frac{1}{2} \Delta \boldsymbol{q}^{(k)T} \boldsymbol{W} \Delta \boldsymbol{q}^{(k)} + \boldsymbol{v}^T \Delta \boldsymbol{q}^{(k)} \tag{3.1}$$

$$\text{s.t.} \;\; \boldsymbol{A} \Delta \boldsymbol{q}^{(k)} = \boldsymbol{b} \tag{3.2}$$

$$\boldsymbol{C} \Delta \boldsymbol{q}^{(k)} \geq \boldsymbol{d} \tag{3.3}$$

$$\text{where} \;\; \boldsymbol{W} = \left(\frac{\partial \boldsymbol{e}(\boldsymbol{q}^{(k)})}{\partial \boldsymbol{q}^{(k)}} \right)^T \left(\frac{\partial \boldsymbol{e}(\boldsymbol{q}^{(k)})}{\partial \boldsymbol{q}^{(k)}} \right) + \boldsymbol{W}_{reg} \tag{3.4}$$

$$\boldsymbol{v} = \left(\frac{\partial \boldsymbol{e}(\boldsymbol{q}^{(k)})}{\partial \boldsymbol{q}^{(k)}} \right)^T \boldsymbol{e}(\boldsymbol{q}^{(k)}) + \boldsymbol{v}_{reg} \tag{3.5}$$

and update configuration:

$$\boldsymbol{q}^{(k+1)} = \boldsymbol{q}^{(k)} + \Delta \boldsymbol{q}^{(k)*} \tag{3.6}$$

:iteration [method]

Return iteration index.

:status [method]

Return status of sqp optimization.

3.2 複数解候補を用いた逐次二次計画法

3.2.1 複数解候補を用いた逐次二次計画法の理論

式 (1.4a) の最適化問題に逐次二次計画法などの制約付き非線形最適化手法を適用すると，初期値から勾配方

向に進行して至る局所最適解が得られると考えられる．したがって解は初期値に強く依存する．

式 (1.4a) の代わりに，以下の最適化問題を考える．

$$\min_{\hat{\boldsymbol{q}}} \sum_{i\in\mathcal{I}} \left\{ F(\boldsymbol{q}^{(i)}) + k_{msc} F_{msc}(\hat{\boldsymbol{q}}; i) \right\} \tag{3.7}$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{q}^{(i)} = \bar{\boldsymbol{b}} \quad i \in \mathcal{I} \tag{3.8}$$

$$\boldsymbol{C}\boldsymbol{q}^{(i)} \geq \bar{\boldsymbol{d}} \quad i \in \mathcal{I} \tag{3.9}$$

$$\text{where} \quad \hat{\boldsymbol{q}} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{q}^{(1)T} & \boldsymbol{q}^{(2)T} & \cdots & \boldsymbol{q}^{(N_{msc})T} \end{pmatrix}^T \tag{3.10}$$

$$\mathcal{I} \stackrel{\text{def}}{=} \{1, 2, \cdots, N_{msc}\} \tag{3.11}$$

$$F_{msc}(\hat{\boldsymbol{q}}; i) \stackrel{\text{def}}{=} -\frac{1}{2} \sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \log \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2 \tag{3.12}$$

$$\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) \stackrel{\text{def}}{=} \boldsymbol{q}^{(i)} - \boldsymbol{q}^{(j)} \tag{3.13}$$

N_{msc} は解候補の個数で，事前に与えるものとする．msc は複数解候補 (multiple solution candidates) を表す．これは，複数の解候補を同時に探索し，それぞれの解候補 $\boldsymbol{q}^{(i)}$ が本来の目的関数 $F(\boldsymbol{q}^{(i)})$ を小さくして，なおかつ，解候補どうしの距離が大きくなるように最適化することを表している．これにより，初期値に依存した唯一の局所解だけでなく，そこから離れた複数の局所解を得ることが可能となり，通常の最適化に比べてより良い解が得られることが期待される．以降では，解候補どうしの距離のコストを表す項 $F_{msc}(\hat{\boldsymbol{q}}; i)$ を解候補分散項と呼ぶ [8]．

解候補分散項のヤコビ行列，ヘッセ行列の各成分は次式で得られる [9]．

$$\begin{aligned}
\nabla_i F_{msc}(\hat{\boldsymbol{q}}; i) &= \frac{\partial F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \boldsymbol{q}^{(i)}} && \text{(3.16a)} \\
&= -\frac{1}{2} \sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \frac{\partial}{\partial \boldsymbol{q}^{(i)}} \log \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2 && \text{(3.16b)} \\
&= -\sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \frac{1}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2} \left(\frac{\partial \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})}{\partial \boldsymbol{q}^{(i)}} \right)^T \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) && \text{(3.16c)} \\
&= -\sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \frac{\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2} && \text{(3.16d)} \\
& && \text{(3.16e)}
\end{aligned}$$

---

[8] 解分散項の log を無くすことは適切ではない．なぜなら，$d = \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|$ として，解分散項の勾配は，

$$\frac{\partial}{\partial d}\left(-\frac{1}{2}\log d^2\right) = -\frac{1}{d} \to -\infty \ \ (d \to +0) \qquad \frac{\partial}{\partial d}\left(-\frac{1}{2}\log d^2\right) = -\frac{1}{d} \to 0 \ \ (d \to \infty) \tag{3.14}$$

となり，最適化により，コンフィギュレーションが近いときほど離れるように更新し，遠くなるとその影響が小さくなる効果が期待される．それに対し，log がない場合の勾配は，

$$\frac{\partial}{\partial d}\left(-\frac{1}{2}d^2\right) = -d \to 0 \ \ (d \to +0) \qquad \frac{\partial}{\partial d}\left(-\frac{1}{2}d^2\right) = -d \to -\infty \ \ (d \to \infty) \tag{3.15}$$

となり，コンフィギュレーションが遠くなるほど離れるように更新し，近いときはその影響が小さくなる．これは，コンフィギュレーションが一致する勾配ゼロの点と，無限に離れ発散する最適値をもち，これらは最適化において望まない挙動をもたらす．

[9] ヘッセ行列の導出は以下を参考にした．`https://math.stackexchange.com/questions/175263/gradient-and-hessian-of-general-2-norm`

$$
\begin{aligned}
\nabla_k F_{msc}(\hat{\boldsymbol{q}}; i) &= \frac{\partial F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \boldsymbol{q}^{(k)}} \qquad k \in \mathcal{I} \wedge k \neq i & (3.17a)\\
&= -\frac{1}{2} \sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \frac{\partial}{\partial \boldsymbol{q}^{(k)}} \log \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2 & (3.17b)\\
&= -\frac{1}{2} \frac{\partial}{\partial \boldsymbol{q}^{(k)}} \log \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2 & (3.17c)\\
&= -\frac{1}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2} \left(\frac{\partial \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})}{\partial \boldsymbol{q}^{(k)}} \right)^T \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) & (3.17d)\\
&= \frac{\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2} & (3.17e)\\
& & (3.17f)
\end{aligned}
$$

$$
\begin{aligned}
\nabla^2_{ii} F_{msc}(\hat{\boldsymbol{q}}; i) &= \frac{\partial^2 F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \boldsymbol{q}^{(i)2}} & (3.18a)\\
&= -\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \frac{\partial}{\partial \boldsymbol{q}^{(i)}} \left(\left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2 \right\}^{-1} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) \right) & (3.18b)\\
&= -\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \left(-2 \left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2 \right\}^{-2} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})^T + \left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2 \right\}^{-1} \boldsymbol{I} \right) & (3.18c)\\
&= -\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \left(-\frac{2}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^4} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})^T + \frac{1}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2} \boldsymbol{I} \right) & (3.18d)\\
&= -\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \boldsymbol{H}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) & (3.18e)
\end{aligned}
$$

ただし,

$$
\boldsymbol{H}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) \stackrel{\mathrm{def}}{=} -\frac{2}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^4} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})^T + \frac{1}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2} \boldsymbol{I} \tag{3.19}
$$

$$
\begin{aligned}
\nabla^2_{ik} F_{msc}(\hat{\boldsymbol{q}}; i) &= \frac{\partial^2 F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \boldsymbol{q}^{(i)} \partial \boldsymbol{q}^{(k)}} \qquad k \in \mathcal{I} \wedge k \neq i & (3.20a)\\
&= -\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \frac{\partial}{\partial \boldsymbol{q}^{(k)}} \left(\left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2 \right\}^{-1} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) \right) & (3.20b)\\
&= -\frac{\partial}{\partial \boldsymbol{q}^{(k)}} \left(\left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2 \right\}^{-1} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) \right) & (3.20c)\\
&= -\left(2 \left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2 \right\}^{-2} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})^T - \left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2 \right\}^{-1} \boldsymbol{I} \right) & (3.20d)\\
&= -\frac{2}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^4} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})^T + \frac{1}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2} \boldsymbol{I} & (3.20e)\\
&= \boldsymbol{H}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) & (3.20f)
\end{aligned}
$$

$$\begin{aligned}
\nabla^2_{kk} F_{msc}(\hat{\boldsymbol{q}}; i) &= \frac{\partial^2 F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \boldsymbol{q}^{(k)2}} \quad k \in \mathcal{I} \wedge k \neq i & (3.21a)\\
&= \frac{\partial}{\partial \boldsymbol{q}^{(k)}} \left(\left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2 \right\}^{-1} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) \right) & (3.21b)\\
&= -\left(-\frac{2}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^4} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})^T + \frac{1}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2} \boldsymbol{I} \right) & (3.21c)\\
&= -\boldsymbol{H}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) & (3.21d)
\end{aligned}$$

$$\begin{aligned}
\nabla^2_{kl} F_{msc}(\hat{\boldsymbol{q}}; i) &= \frac{\partial^2 F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \boldsymbol{q}^{(k)} \partial \boldsymbol{q}^{(l)}} \quad k \in \mathcal{I} \wedge l \in \mathcal{I} \wedge k \neq i \wedge l \neq i \wedge k \neq l & (3.22a)\\
&= \frac{\partial}{\partial \boldsymbol{q}^{(l)}} \left(\left\{ \|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)})\|^2 \right\}^{-1} \boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(k)}) \right) & (3.22b)\\
&= \boldsymbol{O} & (3.22c)
\end{aligned}$$

したがって，解候補分散項のヤコビ行列，ヘッセ行列は次式で表される．

$$\begin{aligned}
\nabla F_{msc}(\hat{\boldsymbol{q}}; i) &= \frac{\partial F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \hat{\boldsymbol{q}}} & (3.23a)\\
&= \begin{pmatrix} \frac{\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(1)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(1)})\|^2} \\ \vdots \\ \frac{\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(i-1)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(i-1)})\|^2} \\ -\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \frac{\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})\|^2} \\ \frac{\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(i+1)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(i+1)})\|^2} \\ \vdots \\ \frac{\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(N_{msc})})}{\|\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(N_{msc})})\|^2} \end{pmatrix} & (3.23b)\\
\boldsymbol{v}_{msc} &\stackrel{\mathrm{def}}{=} \sum_{i \in \mathcal{I}} \nabla F_{msc}(\hat{\boldsymbol{q}}; i) & (3.23c)\\
&= 2 \begin{pmatrix} -\sum_{\substack{j \in \mathcal{I} \\ j \neq 1}} \frac{\boldsymbol{d}(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(j)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(j)})\|^2} \\ \vdots \\ -\sum_{\substack{j \in \mathcal{I} \\ j \neq N_{msc}}} \frac{\boldsymbol{d}(\boldsymbol{q}^{(N_{msc})}, \boldsymbol{q}^{(j)})}{\|\boldsymbol{d}(\boldsymbol{q}^{(N_{msc})}, \boldsymbol{q}^{(j)})\|^2} \end{pmatrix} & (3.23d)
\end{aligned}$$

$$
\nabla^2 F_{msc}(\hat{\boldsymbol{q}}; i) = \frac{\partial^2 F_{msc}(\hat{\boldsymbol{q}}; i)}{\partial \hat{\boldsymbol{q}}^2} \tag{3.24a}
$$

$$
= \begin{array}{c|c}
 & \begin{array}{ccccccc} 1 & \cdots & i-1 & i & i+1 & \cdots & N_{msc} \end{array} \\
\begin{array}{c} 1 \\ \vdots \\ i-1 \\ i \\ i+1 \\ \vdots \\ N_{msc} \end{array} &
\begin{pmatrix}
-\boldsymbol{H}_{i,1} & & & \boldsymbol{H}_{i,1} & & & \\
 & \ddots & & \vdots & & & \\
 & & -\boldsymbol{H}_{i,i-1} & \boldsymbol{H}_{i,i-1} & & & \\
\boldsymbol{H}_{i,1} & \cdots & \boldsymbol{H}_{i,i-1} & -\sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \boldsymbol{H}_{i,j} & \boldsymbol{H}_{i,i+1} & \cdots & \boldsymbol{H}_{i,N_{msc}} \\
 & & & \boldsymbol{H}_{i,i+1} & -\boldsymbol{H}_{i,i+1} & & \\
 & & & \vdots & & \ddots & \\
 & & & \boldsymbol{H}_{i,N_{msc}} & & & -\boldsymbol{H}_{i,N_{msc}}
\end{pmatrix}
\end{array} \tag{3.24b}
$$

$$
\boldsymbol{W}_{msc} \stackrel{\text{def}}{=} \sum_{i\in\mathcal{I}} \nabla^2 F_{msc}(\hat{\boldsymbol{q}}; i) \tag{3.24c}
$$

$$
= 2 \begin{pmatrix}
-\sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \boldsymbol{H}_{1,j} & \boldsymbol{H}_{1,2} & \cdots & \boldsymbol{H}_{1,N_{msc}} \\
\boldsymbol{H}_{2,1} & -\sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \boldsymbol{H}_{2,j} & & \boldsymbol{H}_{2,N_{msc}} \\
\vdots & & \ddots & \vdots \\
\boldsymbol{H}_{N_{msc},1} & \boldsymbol{H}_{N_{msc},2} & \cdots & -\sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \boldsymbol{H}_{N_{msc},j}
\end{pmatrix} \tag{3.24d}
$$

ただし，$\boldsymbol{H}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)})$ を $\boldsymbol{H}_{i,j}$ と略して記す．また，$\boldsymbol{d}(\boldsymbol{q}^{(i)}, \boldsymbol{q}^{(j)}) = -\boldsymbol{d}(\boldsymbol{q}^{(j)}, \boldsymbol{q}^{(i)})$，$\boldsymbol{H}_{i,j} = \boldsymbol{H}_{j,i}$ を利用した．

解候補分散項 $\sum_{i\in\mathcal{I}} F_{msc}(\hat{\boldsymbol{q}}; i)$ による二次計画問題の目的関数 (式 (1.5a)) は次式で表される．

$$
\sum_{i\in\mathcal{I}} \left\{ F_{msc}(\hat{\boldsymbol{q}}_k; i) + \nabla F_{msc}(\hat{\boldsymbol{q}}_k; i)^T \Delta\hat{\boldsymbol{q}}_k + \frac{1}{2} \Delta\hat{\boldsymbol{q}}_k^T \nabla^2 F_{msc}(\hat{\boldsymbol{q}}_k; i) \Delta\hat{\boldsymbol{q}}_k \right\} \tag{3.25}
$$

$$
= \sum_{i\in\mathcal{I}} F_{msc}(\hat{\boldsymbol{q}}_k; i) + \left\{ \sum_{i\in\mathcal{I}} \nabla F_{msc}(\hat{\boldsymbol{q}}_k; i) \right\}^T \Delta\hat{\boldsymbol{q}}_k + \frac{1}{2} \Delta\hat{\boldsymbol{q}}_k^T \left\{ \sum_{i\in\mathcal{I}} \nabla^2 F_{msc}(\hat{\boldsymbol{q}}_k; i) \right\} \Delta\hat{\boldsymbol{q}}_k \tag{3.26}
$$

$$
= \sum_{i\in\mathcal{I}} F_{msc}(\hat{\boldsymbol{q}}_k; i) + \boldsymbol{v}_{msc}^T \Delta\hat{\boldsymbol{q}}_k + \frac{1}{2} \Delta\hat{\boldsymbol{q}}_k^T \boldsymbol{W}_{msc} \Delta\hat{\boldsymbol{q}}_k \tag{3.27}
$$

$\boldsymbol{W}_{msc}$ が必ずしも半正定値行列ではないことに注意する必要がある．以下のようにして $\boldsymbol{W}_{msc}$ に近い正定値行列を計算し用いることで対処する [10]．$\boldsymbol{W}_{msc}$ が次式のように固有値分解されるとする．

$$
\boldsymbol{W}_{msc} = \boldsymbol{V}_{msc} \boldsymbol{D}_{msc} \boldsymbol{V}_{msc}^{-1} \tag{3.28}
$$

ただし，$\boldsymbol{D}_{msc}$ は固有値を対角成分にもつ対角行列，$\boldsymbol{V}_{msc}$ は固有ベクトルを並べた行列である．このとき $\boldsymbol{W}_{msc}$ に近い正定値行列 $\tilde{\boldsymbol{W}}_{msc}$ は次式で得られる．

$$
\tilde{\boldsymbol{W}}_{msc} = \boldsymbol{V}_{msc} \boldsymbol{D}_{msc}^{+} \boldsymbol{V}_{msc}^{-1} \tag{3.29}
$$

ただし，$\boldsymbol{D}_{msc}^{+}$ は $\boldsymbol{D}_{msc}$ の対角成分のうち，負のものを 0 で置き換えた対角行列である．

式 (3.7) において，解候補を分散させながら，最終的に本来の目的関数を最小にする解を得るために，SQP のイテレーションごとに，解候補分散項のスケール k_{msc} を次式のように更新することが有効である．

$$
k_{msc} \leftarrow \min(\gamma_{msc} k_{msc}, k_{msc\text{-}min}) \tag{3.30}
$$

γ_{msc} は $0 < \gamma_{msc} < 1$ なるスケール減少率，$k_{msc\text{-}min}$ はスケール最小値を表す．

[10] $\boldsymbol{W}_{msc}$ が対称行列であることから，以下を参考にした．https://math.stackexchange.com/questions/648809/how-to-find-closest-positive-definite-matrix-of-non-symmetric-matrix#comment1689831_649522

3.2.2 複数解候補を用いた逐次二次計画法の実装

sqp-msc-optimization [class]

:super **sqp-optimization**

:slots (_num-msc number of multiple solution candidates N_{msc})
(_config-task-list list of configuration-task instance)
(_dispersion-scale k_{msc})
(_dispersion-scale-min $k_{msc\text{-}min}$, minimum of k_{msc})
(_dispersion-scale-decrease-ratio γ_{msc}, decrease ration of k_{msc})
(_config-vector-dist2-min minimum squared distance of configuration vector)
(_dispersion-matrix buffer for $\boldsymbol{W}_{msc}$)

複数回候補を用いた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 $\|\boldsymbol{e}(\boldsymbol{q})\|^2$ を最小にするコンフィギュレーション $\boldsymbol{q}$ を，複数の解候補を同時に考慮しながら反復計算により求める．

:init *&rest args &key (num-msc 3)* [method]
(dispersion-scale 0.01)
(dispersion-scale-min 0.0)
(dispersion-scale-decrease-ratio 0.5)
(config-vector-dist2-min 1.000000e-10)
&allow-other-keys

Initialize instance

:config-task-list [method]

Return list of configuration-task instance

:dispersion-matrix [method]

式 (3.23d) 参照．

return $\boldsymbol{W}_{msc} \in \mathbb{R}^{N_{msc} dim(\boldsymbol{q}) \times N_{msc} dim(\boldsymbol{q})}$

:dispersion-vector [method]

式 (3.24d) 参照．

return $\boldsymbol{v}_{msc} \in \mathbb{R}^{N_{msc} dim(\boldsymbol{q})}$

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 $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{N_C}\}$ 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 *&key (robot)* [method]
(object)
(contact-list)
(root-virtual-mode :6dof)
(root-virtual-joint-class-list)
(root-virtual-joint-axis-list)

Initialize instance

:object *&rest args* [method]

return $\mathcal{O}$

:object-with-root-virtual *&rest 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 $\hat{\boldsymbol{w}}_{obj}$ [N] [Nm])
	(_num-contact-obj $N_{cnt\text{-}obj} := \|\mathcal{T}^{cnt\text{-}trg\text{-}obj}\|$)
	(_num-act-react $N_{act\text{-}react} := \|\mathcal{P}^{act\text{-}react}\|$)
	(_dim-wrench-obj $dim(\hat{\boldsymbol{w}}_{obj}) = 6N_{cnt\text{-}obj}$)
	(_contact-target-coords-obj-list $\mathcal{T}^{cnt\text{-}trg\text{-}obj}$)
	(_contact-constraint-obj-list list of contact-constraint instance for object)
	(_act-react-pair-list $\mathcal{P}^{act\text{-}react}$)

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 $\boldsymbol{q}^{(l)}$ と瞬時タスク関数 $\boldsymbol{e}^{(l)}(\boldsymbol{q}^{(l)})$ 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 $\boldsymbol{x}^{(l)}$ と表されるべきだが，このクラス内の説明では省略して $\boldsymbol{x}$ 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 $\boldsymbol{q}$ の取得・更新，タスク関数 $\boldsymbol{e}(\boldsymbol{q})$ の取得，タスク関数のヤコビ行列 $\frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ の取得，コンフィギュレーションの等式・不等式制約 $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{C}, \boldsymbol{d}$ 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 $\mathcal{T}^{cnt\text{-}trg\text{-}obj}$ 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 $\mathcal{P}^{act\text{-}react}$ 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 $\boldsymbol{q}$ は以下から構成される．

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

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

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

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

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

$\hat{\boldsymbol{w}}$ 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．

$$\hat{\boldsymbol{w}} = \begin{pmatrix} \boldsymbol{w}_1^T & \boldsymbol{w}_2^T & \cdots & \boldsymbol{w}_{N_{cnt}}^T \end{pmatrix}^T \tag{4.1}$$

$$= \begin{pmatrix} \boldsymbol{f}_1^T & \boldsymbol{n}_1^T & \boldsymbol{f}_2^T & \boldsymbol{n}_2^T & \cdots & \boldsymbol{f}_{N_{cnt}}^T & \boldsymbol{n}_{N_{cnt}}^T \end{pmatrix}^T \tag{4.2}$$

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

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

$\boldsymbol{e}^{eom\text{-}trans}(\boldsymbol{q}) \in \mathbb{R}^3$ ロボットの力の釣り合い [N]

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

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

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

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

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

:init *&rest args &key (contact-target-coords-obj-list)* [method]
(contact-constraint-obj-list)
(act-react-pair-list)
&allow-other-keys

Initialize instance

:robot-obj-env [method]

return robot-object-environment instance

:wrench-obj [method]

return $\hat{\boldsymbol{w}}_{obj}$

:num-contact-obj [method]

return $N_{cnt\text{-}obj} := |\mathcal{T}^{cnt\text{-}trg\text{-}obj}|$

:dim-variant-config [method]

$$\begin{aligned} dim(\boldsymbol{q_{var}}) &:= dim(\boldsymbol{\theta}) + dim(\hat{\boldsymbol{w}}) + dim(\hat{\boldsymbol{w}}_{\boldsymbol{obj}}) + dim(\boldsymbol{\tau}) && (4.3) \\ &= N_{var\text{-}joint} + 6N_{cnt} + 6N_{cnt\text{-}obj} + N_{drive\text{-}joint} && (4.4) \end{aligned}$$

return $dim(\boldsymbol{q_{var}})$

:dim-task [method]

$$\begin{aligned} dim(\boldsymbol{e}) &:= dim(\boldsymbol{e}^{kin}) + dim(\boldsymbol{e}^{eom\text{-}trans}) + dim(\boldsymbol{e}^{eom\text{-}rot}) + dim(\boldsymbol{e}^{eom\text{-}trans\text{-}obj}) \\ &\quad + dim(\boldsymbol{e}^{eom\text{-}rot\text{-}obj}) + dim(\boldsymbol{e}^{trq}) + dim(\boldsymbol{e}^{posture}) && (4.5) \\ &= 6N_{kin} + 3 + 3 + 3 + 3 + N_{drive\text{-}joint} + N_{posture\text{-}joint} && (4.6) \end{aligned}$$

return $dim(\boldsymbol{e})$

:variant-config-vector [method]

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

:config-vector [method]

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

:set-wrench-obj *wrench-obj-new &key (relative? nil)* [method]

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

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

Set $\boldsymbol{q_{var}}$.

:contact-target-coords-obj-list [method]

$$T_m^{cnt\text{-}trg\text{-}obj} = \{\boldsymbol{p}_m^{cnt\text{-}trg\text{-}obj}, \boldsymbol{R}_m^{cnt\text{-}trg\text{-}obj}\} \quad (m = 1, 2, \cdots, N_{cnt\text{-}obj}) \tag{4.7}$$

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

:contact-constraint-obj-list [method]

return list of contact-constraint instance for object

:wrench-obj-list [method]

return $\{\boldsymbol{w}_{obj,1}, \boldsymbol{w}_{obj,2}, \cdots, \boldsymbol{w}_{obj,N_{obj}}\}$

:force-obj-list [method]

return $\{\boldsymbol{f}_{obj,1}, \boldsymbol{f}_{obj,2}, \cdots, \boldsymbol{f}_{obj,N_{cnt\text{-}obj}}\}$

:moment-obj-list [method]

return $\{\boldsymbol{n}_{obj,1}, \boldsymbol{n}_{obj,2}, \cdots, \boldsymbol{n}_{obj,N_{cnt\text{-}obj}}\}$

:mg-obj-vec [method]

return $m_{obj}\boldsymbol{g}$

:cog-obj *&key (update? t)* [method]

return $\boldsymbol{p}_{Gobj}(\boldsymbol{q})$

:eom-trans-obj-task-value *&key (update? t)* [method]

$$\begin{aligned}
\boldsymbol{e}^{eom\text{-}trans\text{-}obj}(\boldsymbol{q}) &= \boldsymbol{e}^{eom\text{-}trans\text{-}obj}(\hat{\boldsymbol{w}}_{obj}) &(4.8)\\
&= \sum_{m=1}^{N_{cnt\text{-}obj}} \boldsymbol{f}_{obj,m} - m_{obj}\boldsymbol{g} &(4.9)
\end{aligned}$$

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

:eom-rot-obj-task-value *&key (update? t)* [method]

$$\begin{aligned}
\boldsymbol{e}^{eom\text{-}rot\text{-}obj}(\boldsymbol{q}) &= \boldsymbol{e}^{eom\text{-}rot\text{-}obj}(\boldsymbol{\theta}, \hat{\boldsymbol{w}}_{obj}, \boldsymbol{\phi}) &(4.10)\\
&= \sum_{m=1}^{N_{cnt\text{-}obj}} \left\{ \left(\boldsymbol{p}_m^{cnt\text{-}trg\text{-}obj}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}_{Gobj}(\boldsymbol{\theta}, \boldsymbol{\phi})\right) \times \boldsymbol{f}_{obj,m} + \boldsymbol{n}_{obj,m} \right\} &(4.11)
\end{aligned}$$

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

:task-value *&key (update? t)* [method]

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

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

$$\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}trans\text{-}obj}}{\partial \hat{\boldsymbol{w}}_{obj}} &= \begin{pmatrix} \frac{\partial \boldsymbol{e}^{eom\text{-}trans\text{-}obj}}{\partial \boldsymbol{f}_{obj,1}} & \frac{\partial \boldsymbol{e}^{eom\text{-}trans\text{-}obj}}{\partial \boldsymbol{n}_{obj,1}} & \cdots & \frac{\partial \boldsymbol{e}^{eom\text{-}trans\text{-}obj}}{\partial \boldsymbol{f}_{obj,N_{cnt\text{-}obj}}} & \frac{\partial \boldsymbol{e}^{eom\text{-}trans\text{-}obj}}{\partial \boldsymbol{n}_{obj,N_{cnt\text{-}obj}}} \end{pmatrix} &(4.12)\\
&= \begin{pmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 & \cdots & \boldsymbol{I}_3 & \boldsymbol{O}_3 \end{pmatrix} &(4.13)
\end{aligned}$$

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

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

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\theta}} &= \sum_{m=1}^{N_{cnt\text{-}obj}} \left\{ -[\boldsymbol{f}_{obj,m} \times] \left(\boldsymbol{J}^{cnt\text{-}trg\text{-}obj}_{\theta,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{J}_{Gobj\theta}(\boldsymbol{\theta}, \boldsymbol{\phi}) \right) \right\} && (4.14) \\
&= \left[\left(\sum_{m=1}^{N_{cnt\text{-}obj}} \boldsymbol{f}_{obj,m} \right) \times \right] \boldsymbol{J}_{Gobj\theta}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \sum_{m=1}^{N_{cnt\text{-}obj}} [\boldsymbol{f}_{obj,m} \times] \boldsymbol{J}^{cnt\text{-}trg\text{-}obj}_{\theta,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) && (4.15)
\end{aligned}
$$

$\sum_{m=1}^{N_{cnt\text{-}obj}} \boldsymbol{f}_{obj,m} = m_{obj} \boldsymbol{g}$ 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.

$$
\frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\theta}} = [m_{obj} \boldsymbol{g} \times] \boldsymbol{J}_{Gobj\theta}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \sum_{m=1}^{N_{cnt\text{-}obj}} [\boldsymbol{f}_{obj,m} \times] \boldsymbol{J}^{cnt\text{-}trg\text{-}obj}_{\theta,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) \tag{4.16}
$$

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

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

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \hat{\boldsymbol{w}}_{\boldsymbol{obj}}} &= \begin{pmatrix} \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{f}_{obj,1}} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{n}_{obj,1}} & \cdots & \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{f}_{obj,N_{cnt\text{-}obj}}} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{n}_{obj,N_{cnt\text{-}obj}}} \end{pmatrix} && (4.17) \\
\frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{f}_{obj,m}} &= \left[\left(\boldsymbol{p}^{cnt\text{-}trg\text{-}obj}_{m}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}_{Gobj}(\boldsymbol{\theta}, \boldsymbol{\phi}) \right) \times \right] \quad (m = 1, 2, \cdots, N_{cnt\text{-}obj}) && (4.18) \\
\frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{n}_{obj,m}} &= \boldsymbol{I}_3 \quad (m = 1, 2, \cdots, N_{cnt\text{-}obj}) && (4.19)
\end{aligned}
$$

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

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

$$
\begin{aligned}
\frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\phi}} &= \sum_{m=1}^{N_{cnt\text{-}obj}} \left\{ -[\boldsymbol{f}_{obj,m} \times] \left(\boldsymbol{J}^{cnt\text{-}trg\text{-}obj}_{\phi,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{J}_{Gobj\phi}(\boldsymbol{\theta}, \boldsymbol{\phi}) \right) \right\} && (4.20) \\
&= \left[\left(\sum_{m=1}^{N_{cnt\text{-}obj}} \boldsymbol{f}_{obj,m} \right) \times \right] \boldsymbol{J}_{Gobj\phi}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \sum_{m=1}^{N_{cnt\text{-}obj}} [\boldsymbol{f}_{obj,m} \times] \boldsymbol{J}^{cnt\text{-}trg\text{-}obj}_{\phi,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) && (4.21)
\end{aligned}
$$

$\sum_{m=1}^{N_{cnt\text{-}obj}} \boldsymbol{f}_{obj,m} = m_{obj} \boldsymbol{g}$ 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.

$$
\frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\phi}} = [m_{obj} \boldsymbol{g} \times] \boldsymbol{J}_{Gobj\phi}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \sum_{m=1}^{N_{cnt\text{-}obj}} [\boldsymbol{f}_{obj,m} \times] \boldsymbol{J}^{cnt\text{-}trg\text{-}obj}_{\phi,m}(\boldsymbol{\theta}, \boldsymbol{\phi}) \tag{4.22}
$$

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

:variant-task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{var}} = \begin{array}{c c} & \begin{array}{cccc} N_{var\text{-}joint} & 6N_{cnt} & 6N_{cnt\text{-}obj} & N_{drive\text{-}joint} \end{array} \\ \begin{array}{l} 6N_{kin} \\ 3 \\ 3 \\ 3 \\ 3 \\ N_{drive\text{-}joint} \\ N_{posture\text{-}joint} \end{array} & \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\theta}} & & & \\ & \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \hat{\boldsymbol{w}}} & & \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \hat{\boldsymbol{w}}} & & \\ & & \frac{\partial \boldsymbol{e}^{eom\text{-}trans\text{-}obj}}{\partial \hat{\boldsymbol{w}}_{obj}} & \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\theta}} & & \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \hat{\boldsymbol{w}}_{obj}} & \\ \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \hat{\boldsymbol{w}}} & & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\tau}} \\ \frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{\theta}} & & & \end{pmatrix} \end{array} \tag{4.23}$$

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

:invariant-task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{invar}} = \begin{array}{c c} & N_{invar\text{-}joint} \\ \begin{array}{l} 6N_{kin} \\ 3 \\ 3 \\ 3 \\ 3 \\ N_{drive\text{-}joint} \\ N_{posture\text{-}joint} \end{array} & \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} \\ \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} \\ \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\phi}} \\ \\ \end{pmatrix} \end{array} \tag{4.24}$$

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

:task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{var}} & \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}_{invar}} \end{pmatrix} \tag{4.25}$$

$$= \begin{array}{c c} & \begin{array}{ccccc} N_{var\text{-}joint} & 6N_{cnt} & 6N_{cnt\text{-}obj} & N_{drive\text{-}joint} & N_{invar\text{-}joint} \end{array} \\ \begin{array}{l} 6N_{kin} \\ 3 \\ 3 \\ 3 \\ 3 \\ N_{drive\text{-}joint} \\ N_{posture\text{-}joint} \end{array} & \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\theta}} & & & & \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} \\ & \frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \hat{\boldsymbol{w}}} & & & \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \hat{\boldsymbol{w}}} & & & \frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} \\ & & \frac{\partial \boldsymbol{e}^{eom\text{-}trans\text{-}obj}}{\partial \hat{\boldsymbol{w}}_{obj}} & & \\ \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\theta}} & & \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \hat{\boldsymbol{w}}_{obj}} & & \frac{\partial \boldsymbol{e}^{eom\text{-}rot\text{-}obj}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \hat{\boldsymbol{w}}} & & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\tau}} & \frac{\partial \boldsymbol{e}^{trq}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{\theta}} & & & & \end{pmatrix} \end{array} \tag{4.26}$$

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

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

物体の接触レンチ $\boldsymbol{w}_{obj} \in \mathbb{R}^6$ 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．

$$\boldsymbol{C}_{w_{obj}} \boldsymbol{w}_{obj} \geq \boldsymbol{d}_{w_{obj}} \tag{4.27}$$

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

$$\boldsymbol{C}_{w_{obj},m}(\boldsymbol{w}_{obj,m} + \Delta\boldsymbol{w}_{obj,m}) \geq \boldsymbol{d}_{w_{obj},m} \quad (m = 1, 2, \cdots, N_{cnt\text{-}obj}) \tag{4.28}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{w_{obj},m}\Delta\boldsymbol{w}_{obj,m} \geq \boldsymbol{d}_{w_{obj},m} - \boldsymbol{C}_{w_{obj},m}\boldsymbol{w}_{obj,m} \quad (m = 1, 2, \cdots, N_{cnt\text{-}obj}) \tag{4.29}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C}_{w_{obj},1} & & & \\ & \boldsymbol{C}_{w_{obj},2} & & \\ & & \ddots & \\ & & & \boldsymbol{C}_{w_{obj},N_{cnt\text{-}obj}} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{w}_{obj,1} \\ \Delta\boldsymbol{w}_{obj,2} \\ \vdots \\ \Delta\boldsymbol{w}_{obj,N_{cnt\text{-}obj}} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_{w_{obj},1} - \boldsymbol{C}_{w_{obj},1}\boldsymbol{w}_{obj,1} \\ \boldsymbol{d}_{w_{obj},2} - \boldsymbol{C}_{w_{obj},2}\boldsymbol{w}_{obj,2} \\ \vdots \\ \boldsymbol{d}_{w_{obj},N_{cnt\text{-}obj}} - \boldsymbol{C}_{w_{obj},N_{cnt\text{-}obj}}\boldsymbol{w}_{obj,N_{cnt\text{-}obj}} \end{pmatrix} \tag{4.30}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\hat{w}_{obj}}\Delta\hat{\boldsymbol{w}}_{obj} \geq \boldsymbol{d}_{\hat{w}_{obj}} \tag{4.31}$$

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

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

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

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

$$\begin{cases} \boldsymbol{C}_{\theta}\Delta\boldsymbol{\theta} \geq \boldsymbol{d}_{\theta} \\ \boldsymbol{C}_{\hat{w}}\Delta\hat{\boldsymbol{w}} \geq \boldsymbol{d}_{\hat{w}} \\ \boldsymbol{C}_{\hat{w}_{obj}}\Delta\hat{\boldsymbol{w}}_{\boldsymbol{obj}} \geq \boldsymbol{d}_{\hat{w}_{obj}} \\ \boldsymbol{C}_{\tau}\Delta\boldsymbol{\tau} \geq \boldsymbol{d}_{\tau} \end{cases} \tag{4.32}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C}_{\theta} & & & \\ & \boldsymbol{C}_{\hat{w}} & & \\ & & \boldsymbol{C}_{\hat{w}_{obj}} & \\ & & & \boldsymbol{C}_{\tau} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\theta} \\ \Delta\hat{\boldsymbol{w}} \\ \Delta\hat{\boldsymbol{w}}_{obj} \\ \Delta\boldsymbol{\tau} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_{\theta} \\ \boldsymbol{d}_{\hat{w}} \\ \boldsymbol{d}_{\hat{w}_{obj}} \\ \boldsymbol{d}_{\tau} \end{pmatrix} \tag{4.33}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{var}\Delta\boldsymbol{q}_{var} \geq \boldsymbol{d}_{var} \tag{4.34}$$

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

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

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

:act-react-equality-constraint-matrix *&key (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．

$$\hat{\boldsymbol{w}}_{i(m)} + \hat{\boldsymbol{w}}_{obj,j(m)} = \boldsymbol{0} \quad (m = 1, 2, \cdots, N_{act\text{-}react}) \tag{4.35}$$

$$\Leftrightarrow \quad \boldsymbol{A}_{act\text{-}react,robot,m}\hat{\boldsymbol{w}} + \boldsymbol{A}_{act\text{-}react,obj,m}\hat{\boldsymbol{w}}_{obj} = \boldsymbol{0} \quad (m = 1, 2, \cdots, N_{act\text{-}react}) \tag{4.36}$$

$$\text{where} \quad \boldsymbol{A}_{act\text{-}react,robot,m} = \begin{pmatrix} \boldsymbol{O}_6 & \boldsymbol{O}_6 & \cdots & \overset{i(m)\text{ 番目}}{\boldsymbol{I}_6} & \cdots & \boldsymbol{O}_6 & \boldsymbol{O}_6 \end{pmatrix} \in \mathbb{R}^{6 \times 6N_{cnt}} \tag{4.37}$$

$$\boldsymbol{A}_{act\text{-}react,obj,m} = \begin{pmatrix} \boldsymbol{O}_6 & \boldsymbol{O}_6 & \cdots & \overset{j(m)\text{ 番目}}{\boldsymbol{I}_6} & \cdots & \boldsymbol{O}_6 & \boldsymbol{O}_6 \end{pmatrix} \in \mathbb{R}^{6 \times 6N_{cnt\text{-}obj}} \tag{4.38}$$

$$\Leftrightarrow \quad \boldsymbol{A}_{act\text{-}react,robot}\hat{\boldsymbol{w}} + \boldsymbol{A}_{act\text{-}react,obj}\hat{\boldsymbol{w}}_{obj} = \boldsymbol{0} \tag{4.39}$$

$$\text{where} \quad \boldsymbol{A}_{act\text{-}react,robot} = \begin{pmatrix} \boldsymbol{A}_{act\text{-}react,robot,1} \\ \vdots \\ \boldsymbol{A}_{act\text{-}react,robot,N_{act\text{-}react}} \end{pmatrix} \in \mathbb{R}^{6N_{act\text{-}react} \times 6N_{cnt}} \tag{4.40}$$

$$\boldsymbol{A}_{act\text{-}react,obj} = \begin{pmatrix} \boldsymbol{A}_{act\text{-}react,obj,1} \\ \vdots \\ \boldsymbol{A}_{act\text{-}react,obj,N_{act\text{-}react}} \end{pmatrix} \in \mathbb{R}^{6N_{act\text{-}react} \times 6N_{cnt\text{-}obj}} \tag{4.41}$$

$$\Leftrightarrow \quad \boldsymbol{A}_{act\text{-}react} \begin{pmatrix} \hat{\boldsymbol{w}} \\ \hat{\boldsymbol{w}}_{obj} \end{pmatrix} = \boldsymbol{0} \in \mathbb{R}^{6N_{act\text{-}react}} \tag{4.42}$$

$$\text{where} \quad \boldsymbol{A}_{act\text{-}react} = \begin{pmatrix} \boldsymbol{A}_{act\text{-}react,robot} & \boldsymbol{A}_{act\text{-}react,obj} \end{pmatrix} \in \mathbb{R}^{6N_{act\text{-}react} \times (6N_{cnt} + 6N_{cnt\text{-}obj})} \tag{4.43}$$

$$\Leftrightarrow \quad \boldsymbol{A}_{act\text{-}react} \begin{pmatrix} \hat{\boldsymbol{w}} + \Delta\hat{\boldsymbol{w}} \\ \hat{\boldsymbol{w}}_{obj} + \Delta\hat{\boldsymbol{w}}_{obj} \end{pmatrix} = \boldsymbol{0} \tag{4.44}$$

$$\Leftrightarrow \quad \boldsymbol{A}_{act\text{-}react} \begin{pmatrix} \Delta\hat{\boldsymbol{w}} \\ \Delta\hat{\boldsymbol{w}}_{obj} \end{pmatrix} = \boldsymbol{b}_{act\text{-}react} \tag{4.45}$$

$$\text{where} \quad \boldsymbol{b}_{act\text{-}react} = -\boldsymbol{A}_{act\text{-}react} \begin{pmatrix} \hat{\boldsymbol{w}} \\ \hat{\boldsymbol{w}}_{obj} \end{pmatrix} \tag{4.46}$$

$i(m), j(m)$ 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 $\boldsymbol{A}_{act\text{-}react} \in \mathbb{R}^{6N_{act\text{-}react} \times (6N_{cnt} + 6N_{cnt\text{-}obj})}$

:act-react-equality-constraint-vector *&key (update? t)* [method]

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

:variant-config-equality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{A}_{act\text{-}react} \begin{pmatrix} \Delta\hat{\boldsymbol{w}} \\ \Delta\hat{\boldsymbol{w}}_{obj} \end{pmatrix} = \boldsymbol{b}_{act\text{-}react} \tag{4.47}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{O} & \boldsymbol{A}_{act\text{-}react} & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\theta} \\ \Delta\hat{\boldsymbol{w}} \\ \Delta\hat{\boldsymbol{w}}_{obj} \\ \Delta\boldsymbol{\tau} \end{pmatrix} = \boldsymbol{b}_{act\text{-}react} \tag{4.48}$$

$$\Leftrightarrow \quad \boldsymbol{A}_{var}\Delta\boldsymbol{q}_{var} = \boldsymbol{b}_{var} \tag{4.49}$$

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

:variant-config-equality-constraint-vector *&key (update? t)* [method]

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

:invariant-config-equality-constraint-matrix *&key (update? nil)* [method]

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

:invariant-config-equality-constraint-vector *&key (update? t)* [method]

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

:config-equality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{A}_{var}\Delta\boldsymbol{q}_{var} = \boldsymbol{b}_{var} \tag{4.50}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{A}_{var} & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{q}_{var} \\ \Delta\boldsymbol{q}_{invar} \end{pmatrix} = \boldsymbol{b}_{var} \tag{4.51}$$

$$\Leftrightarrow \quad \boldsymbol{A}\Delta\boldsymbol{q} = \boldsymbol{b} \tag{4.52}$$

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

:config-equality-constraint-vector *&key (update? t)* [method]

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

:torque-regular-matrix *&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．

$$F_{tau}(\boldsymbol{q}) = \left\| \frac{\boldsymbol{\tau}}{\boldsymbol{\tau}_{max}} \right\|^2 \quad \text{(ベクトルの要素ごとの割り算を表す)} \tag{4.53}$$

$$= \boldsymbol{\tau}^T \bar{\boldsymbol{W}}_{trq} \boldsymbol{\tau} \tag{4.54}$$

ここで,

$$\bar{\boldsymbol{W}}_{trq} := \begin{pmatrix} \frac{1}{\tau_{max,1}^2} & & & \\ & \frac{1}{\tau_{max,2}^2} & & \\ & & \ddots & \\ & & & \frac{1}{\tau_{max,N_{drive\text{-}joint}}^2} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{\tau}) \times dim(\boldsymbol{\tau})} \tag{4.55}$$

only-variant? is true:

$$\boldsymbol{W}_{trq} := \begin{array}{c} \\ dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\hat{\boldsymbol{w}}_{obj}) \\ dim(\boldsymbol{\tau}) \end{array} \begin{array}{c} \begin{array}{cccc} dim(\boldsymbol{\theta}) & dim(\hat{\boldsymbol{w}}) & dim(\hat{\boldsymbol{w}}_{obj}) & dim(\boldsymbol{\tau}) \end{array} \\ \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \bar{\boldsymbol{W}}_{trq} \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q}_{var}) \times dim(\boldsymbol{q}_{var})} \tag{4.56}$$

otherwise:

$$\boldsymbol{W}_{trq} := \begin{array}{c} \\ dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\hat{\boldsymbol{w}}_{obj}) \\ dim(\boldsymbol{\tau}) \\ dim(\boldsymbol{\phi}) \end{array} \begin{array}{c} \begin{array}{ccccc} dim(\boldsymbol{\theta}) & dim(\hat{\boldsymbol{w}}) & dim(\hat{\boldsymbol{w}}_{obj}) & dim(\boldsymbol{\tau}) & dim(\boldsymbol{\phi}) \end{array} \\ \begin{pmatrix} & & & & \\ & & & & \\ & & & & \\ & & & \bar{\boldsymbol{W}}_{trq} & \\ & & & & \end{pmatrix} \end{array} \in \mathbb{R}^{dim(\boldsymbol{q}) \times dim(\boldsymbol{q})} \tag{4.57}$$

return $\boldsymbol{W}_{trq}$

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

$$\begin{aligned} \bar{\boldsymbol{v}}_{trq} &:= \bar{\boldsymbol{W}}_{trq}\boldsymbol{\tau} & (4.58)\\ &= \begin{pmatrix} \frac{\tau_1}{\tau_{max,1}^2} \\ \frac{\tau_2}{\tau_{max,2}^2} \\ \vdots \\ \frac{\tau_{dim(\boldsymbol{\tau})}}{\tau_{max,dim(\boldsymbol{\tau})}^2} \end{pmatrix} \in \mathbb{R}^{dim(\boldsymbol{\tau})} & (4.59) \end{aligned}$$

only-variant? is true:

$$\boldsymbol{v}_{trq} := \begin{array}{l} \\ dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\hat{\boldsymbol{w}}_{obj}) \\ dim(\boldsymbol{\tau}) \end{array} \begin{array}{c} 1 \\ \left(\begin{array}{c} \\ \\ \\ \bar{\boldsymbol{v}}_{trq} \end{array}\right) \end{array} \in \mathbb{R}^{dim(\boldsymbol{q}_{var})} \quad (4.60)$$

otherwise:

$$\boldsymbol{v}_{trq} := \begin{array}{l} \\ dim(\boldsymbol{\theta}) \\ dim(\hat{\boldsymbol{w}}) \\ dim(\hat{\boldsymbol{w}}_{obj}) \\ dim(\boldsymbol{\tau}) \\ dim(\boldsymbol{\phi}) \end{array} \begin{array}{c} 1 \\ \left(\begin{array}{c} \\ \\ \\ \bar{\boldsymbol{v}}_{trq} \\ \\ \end{array}\right) \end{array} \in \mathbb{R}^{dim(\boldsymbol{q})} \quad (4.61)$$

return $\boldsymbol{v}_{trq}$

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

$$\boldsymbol{C}_{col} := N_{col} \begin{array}{c} \begin{array}{ccccc} dim(\boldsymbol{\theta}) & dim(\hat{\boldsymbol{w}}) & dim(\hat{\boldsymbol{w}}_{obj}) & dim(\boldsymbol{\tau}) & dim(\boldsymbol{\phi}) \end{array} \\ \begin{pmatrix} \boldsymbol{C}_{col,\theta} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{C}_{col,\phi} \end{pmatrix} \end{array} \quad (4.62)$$

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

: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理論

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．

$$b_{i,0}(t) \overset{\text{def}}{=} \begin{cases} 1 & \text{if} \quad t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.63}$$

$$b_{i,n}(t) \overset{\text{def}}{=} \frac{t - t_i}{t_{i+n} - t_i} b_{i,n-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} b_{i+1,n-1}(t) \tag{4.64}$$

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．

$$t_n = t_s \tag{4.65}$$

$$t_m = t_f \tag{4.66}$$

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

$$t_i = \frac{i-n}{m-n}(t_f - t_s) + t_s \tag{4.67}$$

$$= hi + \frac{mt_s - nt_f}{m-n} \tag{4.68}$$

ただし，

$$h \overset{\text{def}}{=} \frac{t_f - t_s}{m-n} \tag{4.69}$$

式 (4.68) を式 (4.63) , 式 (4.64) に代入すると，B スプライン基底関数は次式で得られる．

$$b_{i,0}(t) = \begin{cases} 1 & \text{if} \quad t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.70}$$

$$b_{i,n}(t) = \frac{(t - t_i) b_{i,n-1}(t) + (t_{i+n+1} - t) b_{i+1,n-1}(t)}{nh} \tag{4.71}$$

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 (4.70) , 式 (4.71) 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．

$$\sum_{i=0}^{m-1} b_{i,n}(t) = 1 \quad (t_s \le t \le t_f) \tag{4.72}$$

$$0 \le b_{i,n}(t) \le 1 \quad (i = 0, 1, \cdots, m-1, \ t_s \le t \le t_f) \tag{4.73}$$

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 [11].

$$\dot{\boldsymbol{b}}_n(t) = \frac{d\boldsymbol{b}_n(t)}{dt} = \boldsymbol{D}\boldsymbol{b}_{n-1}(t) \tag{4.74}$$

ただし,

$$\boldsymbol{b}_n(t) \overset{\text{def}}{=} \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{m-1,n}(t) \end{pmatrix} \in \mathbb{R}^m \tag{4.75}$$

$$\boldsymbol{D} \overset{\text{def}}{=} \frac{1}{h}\begin{pmatrix} 1 & -1 & & & \boldsymbol{O} \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -1 \\ \boldsymbol{O} & & & & 1 \end{pmatrix} \in \mathbb{R}^{m\times m} \tag{4.76}$$

したがって, k 階微分に関して次式が成り立つ.

$$\boldsymbol{b}_n^{(k)}(t) = \frac{d^{(k)}\boldsymbol{b}_n(t)}{dt^{(k)}} = \boldsymbol{D}^k\boldsymbol{b}_{n-k}(t) \tag{4.77}$$

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.

$$\theta_j(t) \overset{\text{def}}{=} \sum_{i=0}^{m-1} p_{j,i}b_{i,n}(t) = \boldsymbol{p}_j^T\boldsymbol{b}_n(t) \in \mathbb{R} \quad (t_s \le t \le t_f) \tag{4.78}$$

ただし,

$$\boldsymbol{p}_j = \begin{pmatrix} p_{j,0} \\ p_{j,1} \\ \vdots \\ p_{j,m-1} \end{pmatrix} \in \mathbb{R}^m, \quad \boldsymbol{b}_n(t) = \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{m-1,n}(t) \end{pmatrix} \in \mathbb{R}^m \tag{4.79}$$

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

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

$$\boldsymbol{\theta}(t) \overset{\text{def}}{=} \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_{N_{joint}}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1^T\boldsymbol{b}_n(t) \\ \boldsymbol{p}_2^T\boldsymbol{b}_n(t) \\ \vdots \\ \boldsymbol{p}_{N_{joint}}^T\boldsymbol{b}_n(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_{N_{joint}}^T \end{pmatrix}\boldsymbol{b}_n(t) = \boldsymbol{P}\boldsymbol{b}_n(t) \in \mathbb{R}^{N_{joint}} \tag{4.80}$$

ただし,

$$\boldsymbol{P} \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_{N_{joint}}^T \end{pmatrix} \in \mathbb{R}^{N_{joint}\times m} \tag{4.81}$$

[11] 数学的帰納法で証明できる. http://mat.fsv.cvut.cz/gcg/sbornik/prochazkova.pdf

式 (4.80) 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．

$$\boldsymbol{\theta}(t) = \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_{N_{joint}}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_n^T(t)\boldsymbol{p}_1 \\ \boldsymbol{b}_n^T(t)\boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{b}_n^T(t)\boldsymbol{p}_{N_{joint}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_n^T(t) & & & \boldsymbol{O} \\ & \boldsymbol{b}_n^T(t) & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{b}_n^T(t) \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_{N_{joint}} \end{pmatrix} = \boldsymbol{B}_n(t)\boldsymbol{p} \in \mathbb{R}^{N_{joint}} \tag{4.82}$$

ただし，

$$\boldsymbol{B}_n(t) \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{b}_n^T(t) & & & \boldsymbol{O} \\ & \boldsymbol{b}_n^T(t) & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{b}_n^T(t) \end{pmatrix} \in \mathbb{R}^{N_{joint} \times mN_{joint}}, \quad \boldsymbol{p} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_{N_{joint}} \end{pmatrix} \in \mathbb{R}^{mN_{joint}} \tag{4.83}$$

B スプラインによる関節角軌道の微分

式 (4.80) と式 (4.74) から，関節角速度軌道は次式で得られる．

$$\begin{aligned}
\dot{\boldsymbol{\theta}}(t) &= \boldsymbol{P}\dot{\boldsymbol{b}}_n(t) && (4.84) \\
&= \boldsymbol{P}\boldsymbol{D}\boldsymbol{b}_{n-1}(t) && (4.85) \\
&= \begin{pmatrix} \boldsymbol{p}_1^T \\ \vdots \\ \boldsymbol{p}_{N_{joint}}^T \end{pmatrix} \boldsymbol{D}\boldsymbol{b}_{n-1}(t) && (4.86) \\
&= \begin{pmatrix} \boldsymbol{p}_1^T\boldsymbol{D}\boldsymbol{b}_{n-1}(t) \\ \vdots \\ \boldsymbol{p}_{N_{joint}}^T\boldsymbol{D}\boldsymbol{b}_{n-1}(t) \end{pmatrix} && (4.87) \\
&= \begin{pmatrix} \boldsymbol{b}_{n-1}^T(t)\boldsymbol{D}^T\boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{b}_{n-1}^T(t)\boldsymbol{D}^T\boldsymbol{p}_{N_{joint}} \end{pmatrix} && (4.88) \\
&= \begin{pmatrix} \boldsymbol{b}_{n-1}^T(t)\boldsymbol{D}^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \boldsymbol{b}_{n-1}^T(t)\boldsymbol{D}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_{N_{joint}} \end{pmatrix} && (4.89) \\
&= \begin{pmatrix} \boldsymbol{b}_{n-1}^T(t) & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \boldsymbol{b}_{n-1}^T(t) \end{pmatrix} \begin{pmatrix} \boldsymbol{D}^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \boldsymbol{D}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_{N_{joint}} \end{pmatrix} && (4.90) \\
&= \boldsymbol{B}_{n-1}(t)\hat{\boldsymbol{D}}_1\boldsymbol{p} && (4.91)
\end{aligned}$$

ただし，

$$\hat{\boldsymbol{D}}_1 = \begin{pmatrix} \boldsymbol{D}^T & & & \boldsymbol{O} \\ & \boldsymbol{D}^T & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{D}^T \end{pmatrix} \in \mathbb{R}^{mN_{joint} \times mN_{joint}} \tag{4.92}$$

同様にして，関節角軌道の k 階微分は次式で得られる．

$$\boldsymbol{\theta}^{(k)}(t) = \frac{d^{(k)}\boldsymbol{\theta}(t)}{dt^{(k)}} \tag{4.93}$$

$$= \boldsymbol{P}\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t) \tag{4.94}$$

$$= \boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\boldsymbol{p} \tag{4.95}$$

ただし，

$$\hat{\boldsymbol{D}}_k = \begin{pmatrix} (\boldsymbol{D}^k)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & (\boldsymbol{D}^k)^T \end{pmatrix} = (\hat{\boldsymbol{D}}_1)^k \in \mathbb{R}^{mN_{joint}\times mN_{joint}} \tag{4.96}$$

計算時間は式 (4.94) のほうが式 (4.95) 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．

$$\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{\theta}) \tag{4.97}$$

関節角軌道が式 (4.82) 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．

$$\boldsymbol{r}(t) = \boldsymbol{f}(\boldsymbol{\theta}(t)) = \boldsymbol{f}(\boldsymbol{B}_n(t)\boldsymbol{p}) \tag{4.98}$$

$l = 1, 2, \cdots, N_{tm}$ 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．

$$\boldsymbol{e}(\boldsymbol{p},\boldsymbol{t}) \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{e}_1(\boldsymbol{p},\boldsymbol{t}) \\ \boldsymbol{e}_2(\boldsymbol{p},\boldsymbol{t}) \\ \vdots \\ \boldsymbol{e}_{N_{tm}}(\boldsymbol{p},\boldsymbol{t}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_1 - \boldsymbol{f}(\boldsymbol{\theta}(t_1)) \\ \boldsymbol{r}_2 - \boldsymbol{f}(\boldsymbol{\theta}(t_2)) \\ \vdots \\ \boldsymbol{r}_{N_{tm}} - \boldsymbol{f}(\boldsymbol{\theta}(t_{N_{tm}})) \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_1 - \boldsymbol{f}(\boldsymbol{B}_n(t_1)\boldsymbol{p}) \\ \boldsymbol{r}_2 - \boldsymbol{f}(\boldsymbol{B}_n(t_2)\boldsymbol{p}) \\ \vdots \\ \boldsymbol{r}_{N_{tm}} - \boldsymbol{f}(\boldsymbol{B}_n(t_{N_{tm}})\boldsymbol{p}) \end{pmatrix} \in \mathbb{R}^{6N_{tm}} \tag{4.99}$$

ただし，

$$\boldsymbol{e}_l(\boldsymbol{p},\boldsymbol{t}) \overset{\text{def}}{=} \boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{\theta}(t_l)) = \boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{B}_n(t_l)\boldsymbol{p}) \in \mathbb{R}^6 \ (l = 1, 2, \cdots, N_{tm}) \tag{4.100}$$

$$\boldsymbol{t} \overset{\text{def}}{=} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_{N_{tm}} \end{pmatrix} \in \mathbb{R}^{N_{tm}} \tag{4.101}$$

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．

$$F(\boldsymbol{p},\boldsymbol{t}) \overset{\text{def}}{=} \frac{1}{2}\|\boldsymbol{e}(\boldsymbol{p},\boldsymbol{t})\|^2 \tag{4.102}$$

$$= \frac{1}{2}\sum_{l=1}^{N_{tm}} \|\boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{\theta}(t_l))\|^2 \tag{4.103}$$

$$= \frac{1}{2}\sum_{l=1}^{N_{tm}} \|\boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{B}_n(t_l)\boldsymbol{p})\|^2 \tag{4.104}$$

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

$$\tilde{e}_{l,i}(\boldsymbol{p}, \boldsymbol{t}) \stackrel{\mathrm{def}}{=} \begin{cases} e_{l,i}(\boldsymbol{p}, \boldsymbol{t}) - e_{tol,l,i} & e_{l,i}(\boldsymbol{p}, \boldsymbol{t}) > e_{tol,l,i} \\ e_{l,i}(\boldsymbol{p}, \boldsymbol{t}) + e_{tol,l,i} & e_{l,i}(\boldsymbol{p}, \boldsymbol{t}) < -e_{tol,l,i} \\ 0 & \text{otherwise} \end{cases} \quad (i = 1, 2, \cdots, 6) \tag{4.105}$$

$\tilde{e}_{l,i}(\boldsymbol{p}, \boldsymbol{t})$ は $\tilde{\boldsymbol{e}}_l(\boldsymbol{p}, \boldsymbol{t})$ の i 番目の要素である．$e_{l,i}(\boldsymbol{p}, \boldsymbol{t})$ は $\boldsymbol{e}(\boldsymbol{p}, \boldsymbol{t})$ 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 (4.104) 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 (4.99) 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 $\boldsymbol{e}_l(\boldsymbol{p}, \boldsymbol{t})$ の制御点 $\boldsymbol{p}$ に対するヤコビ行列は次式で求められる．

$$\begin{aligned}
\frac{\partial \boldsymbol{e}_l(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{p}} &= \frac{\partial}{\partial \boldsymbol{p}} \{\boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{B}_n(t_l)\boldsymbol{p})\} && (4.106) \\
&= -\frac{\partial}{\partial \boldsymbol{p}} \boldsymbol{f}(\boldsymbol{B}_n(t_l)\boldsymbol{p}) && (4.107) \\
&= -\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}(t_l)} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{p}} && (4.108) \\
&= -\boldsymbol{J}(\boldsymbol{\theta}(t_l)) \frac{\partial}{\partial \boldsymbol{p}} \{\boldsymbol{B}_n(t_l)\boldsymbol{p}\} && (4.109) \\
&= -\boldsymbol{J}(\boldsymbol{\theta}(t_l)) \boldsymbol{B}_n(t_l) && (4.110)
\end{aligned}$$

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

$$\boldsymbol{J} \stackrel{\mathrm{def}}{=} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} \tag{4.111}$$

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 $\boldsymbol{e}_l(\boldsymbol{p}, \boldsymbol{t})$ のタイミング $\boldsymbol{t}$ に対するヤコビ行列は次式で求められる．

$$\begin{aligned}
\frac{\partial \boldsymbol{e}_l(\boldsymbol{p}, \boldsymbol{t})}{\partial t_l} &= \frac{\partial}{\partial t_l} \{\boldsymbol{r}_l - \boldsymbol{f}(\boldsymbol{P}\boldsymbol{b}_n(t_l))\} && (4.112) \\
&= -\frac{\partial}{\partial t_l} \boldsymbol{f}(\boldsymbol{P}\boldsymbol{b}_n(t_l)) && (4.113) \\
&= -\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}(t_l)} \frac{\partial \boldsymbol{\theta}}{\partial t_l} && (4.114) \\
&= -\boldsymbol{J}(\boldsymbol{\theta}(t_l)) \frac{\partial}{\partial t_l} \{\boldsymbol{P}\boldsymbol{b}_n(t_l)\} && (4.115) \\
&= -\boldsymbol{J}(\boldsymbol{\theta}(t_l)) \boldsymbol{P}\dot{\boldsymbol{b}}_n(t_l) && (4.116) \\
&= -\boldsymbol{J}(\boldsymbol{\theta}(t_l)) \boldsymbol{P}\boldsymbol{D}\boldsymbol{b}_{n-1}(t_l) && (4.117)
\end{aligned}$$

途中の変形で，$\boldsymbol{\theta}(\boldsymbol{p}; t) = \boldsymbol{P}\boldsymbol{b}_n(t)$ 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．

$$\begin{aligned} \boldsymbol{e}_{sv}(\boldsymbol{p},\boldsymbol{t}) &\stackrel{\text{def}}{=} \dot{\boldsymbol{\theta}}(t_s) && (4.118)\\ &= \boldsymbol{B}_{n-1}(t_s)\hat{\boldsymbol{D}}_1\boldsymbol{p} && (4.119)\\ &= \boldsymbol{P}\boldsymbol{D}\boldsymbol{b}_{n-1}(t_s) && (4.120)\\ \boldsymbol{e}_{fv}(\boldsymbol{p},\boldsymbol{t}) &\stackrel{\text{def}}{=} \dot{\boldsymbol{\theta}}(t_f) && (4.121)\\ &= \boldsymbol{B}_{n-1}(t_f)\hat{\boldsymbol{D}}_1\boldsymbol{p} && (4.122)\\ &= \boldsymbol{P}\boldsymbol{D}\boldsymbol{b}_{n-1}(t_f) && (4.123)\\ \boldsymbol{e}_{sa}(\boldsymbol{p},\boldsymbol{t}) &\stackrel{\text{def}}{=} \ddot{\boldsymbol{\theta}}(t_s) && (4.124)\\ &= \boldsymbol{B}_{n-2}(t_s)\hat{\boldsymbol{D}}_2\boldsymbol{p} && (4.125)\\ &= \boldsymbol{P}\boldsymbol{D}^2\boldsymbol{b}_{n-2}(t_s) && (4.126)\\ \boldsymbol{e}_{fa}(\boldsymbol{p},\boldsymbol{t}) &\stackrel{\text{def}}{=} \ddot{\boldsymbol{\theta}}(t_f) && (4.127)\\ &= \boldsymbol{B}_{n-2}(t_f)\hat{\boldsymbol{D}}_2\boldsymbol{p} && (4.128)\\ &= \boldsymbol{P}\boldsymbol{D}^2\boldsymbol{b}_{n-2}(t_f) && (4.129) \end{aligned}$$

制御点で微分したヤコビ行列は次式で表される．

$$\begin{aligned} \frac{\partial \boldsymbol{e}_{sv}(\boldsymbol{p},\boldsymbol{t})}{\partial \boldsymbol{p}} &= \boldsymbol{B}_{n-1}(t_s)\hat{\boldsymbol{D}}_1 && (4.130)\\ \frac{\partial \boldsymbol{e}_{fv}(\boldsymbol{p},\boldsymbol{t})}{\partial \boldsymbol{p}} &= \boldsymbol{B}_{n-1}(t_f)\hat{\boldsymbol{D}}_1 && (4.131)\\ \frac{\partial \boldsymbol{e}_{sa}(\boldsymbol{p},\boldsymbol{t})}{\partial \boldsymbol{p}} &= \boldsymbol{B}_{n-2}(t_s)\hat{\boldsymbol{D}}_2 && (4.132)\\ \frac{\partial \boldsymbol{e}_{fa}(\boldsymbol{p},\boldsymbol{t})}{\partial \boldsymbol{p}} &= \boldsymbol{B}_{n-2}(t_f)\hat{\boldsymbol{D}}_2 && (4.133) \end{aligned}$$

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．

$$\begin{aligned} \frac{\partial \boldsymbol{e}_{sv}(\boldsymbol{p},\boldsymbol{t})}{\partial t_s} &= \boldsymbol{P}\boldsymbol{D}\frac{\partial \boldsymbol{b}_{n-1}(t_s)}{\partial t_s} = \boldsymbol{P}\boldsymbol{D}^2\boldsymbol{b}_{n-2}(t_s) && (4.134)\\ \frac{\partial \boldsymbol{e}_{fv}(\boldsymbol{p},\boldsymbol{t})}{\partial t_f} &= \boldsymbol{P}\boldsymbol{D}\frac{\partial \boldsymbol{b}_{n-1}(t_f)}{\partial t_f} = \boldsymbol{P}\boldsymbol{D}^2\boldsymbol{b}_{n-2}(t_f) && (4.135)\\ \frac{\partial \boldsymbol{e}_{sa}(\boldsymbol{p},\boldsymbol{t})}{\partial t_s} &= \boldsymbol{P}\boldsymbol{D}^2\frac{\partial \boldsymbol{b}_{n-2}(t_s)}{\partial t_s} = \boldsymbol{P}\boldsymbol{D}^3\boldsymbol{b}_{n-3}(t_s) && (4.136)\\ \frac{\partial \boldsymbol{e}_{fa}(\boldsymbol{p},\boldsymbol{t})}{\partial t_f} &= \boldsymbol{P}\boldsymbol{D}^2\frac{\partial \boldsymbol{b}_{n-2}(t_f)}{\partial t_f} = \boldsymbol{P}\boldsymbol{D}^3\boldsymbol{b}_{n-3}(t_f) && (4.137) \end{aligned}$$

関節角上下限制約

式 (4.78) の関節角軌道定義において，

$$\boldsymbol{p}_j \leq \theta_{max,j}\boldsymbol{1}_m \tag{4.138}$$

のとき，B スプラインの凸包性 (式 (4.72) , 式 (4.73)) より次式が成り立つ．ただし，$\boldsymbol{1}_m \in \mathbb{R}^m$ は全要素が 1

の m 次元ベクトルである．

$$\begin{aligned}\theta_j(t) &= \sum_{i=0}^{m-1} p_{j,i} b_{i,n}(t) && (4.139)\\ &\leq \sum_{i=0}^{m-1} \theta_{max,j} b_{i,n}(t) && (4.140)\\ &= \theta_{max,j} \sum_{i=0}^{m-1} b_{i,n}(t) && (4.141)\\ &= \theta_{max,j} && (4.142)\end{aligned}$$

同様に，$\theta_{min,j}\mathbf{1}_m \leq \boldsymbol{p}_j$ 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．

$$\theta_{min,j}\mathbf{1}_m \leq \boldsymbol{p}_j \leq \theta_{max,j}\mathbf{1}_m \ (j = 1, 2, \cdots, N_{joint}) \tag{4.143}$$

つまり，

$$\begin{aligned} & \hat{\boldsymbol{E}}\boldsymbol{\theta}_{min} \leq \boldsymbol{p} \leq \hat{\boldsymbol{E}}\boldsymbol{\theta}_{max} && (4.144)\\ \Leftrightarrow \quad & \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \boldsymbol{p} \geq \begin{pmatrix} \hat{\boldsymbol{E}}\boldsymbol{\theta}_{min} \\ -\hat{\boldsymbol{E}}\boldsymbol{\theta}_{max} \end{pmatrix} && (4.145)\end{aligned}$$

ただし，

$$\hat{\boldsymbol{E}} \stackrel{\mathrm{def}}{=} \begin{pmatrix} \mathbf{1}_m & & & \mathbf{0}_m \\ & \mathbf{1}_m & & \\ & & \ddots & \\ \mathbf{0}_m & & & \mathbf{1}_m \end{pmatrix} \in \mathbb{R}^{mN_{joint} \times N_{joint}} \tag{4.146}$$

これは，逐次二次計画法の中で，次式の不等式制約となる．

$$\begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{p} \geq \begin{pmatrix} \hat{\boldsymbol{E}}\boldsymbol{\theta}_{min} - \boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{\theta}_{max} + \boldsymbol{p} \end{pmatrix} \tag{4.147}$$

関節角速度・角加速度上下限制約

式 (4.78) と式 (4.74) より，関節角速度軌道，角加速度軌道は次式で表される．

$$\begin{aligned}\dot{\theta}_j(t) &= \boldsymbol{p}_j^T \dot{\boldsymbol{b}}_n(t) = \boldsymbol{p}_j^T \boldsymbol{D}\boldsymbol{b}_{n-1}(t) = (\boldsymbol{D}^T\boldsymbol{p}_j)^T \boldsymbol{b}_{n-1}(t) \in \mathbb{R} \ \ (t_s \leq t \leq t_f) && (4.148)\\ \ddot{\theta}_j(t) &= \boldsymbol{p}_j^T \ddot{\boldsymbol{b}}_n(t) = \boldsymbol{p}_j^T \boldsymbol{D}^2\boldsymbol{b}_{n-2}(t) = ((\boldsymbol{D}^2)^T\boldsymbol{p}_j)^T \boldsymbol{b}_{n-2}(t) \in \mathbb{R} \ \ (t_s \leq t \leq t_f) && (4.149)\end{aligned}$$

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．

$$\begin{aligned} -v_{max,j}\mathbf{1}_m \leq \boldsymbol{D}^T\boldsymbol{p}_j \leq v_{max,j}\mathbf{1}_m \ (j = 1, 2, \cdots, N_{joint}) && (4.150)\\ -a_{max,j}\mathbf{1}_m \leq (\boldsymbol{D}^2)^T\boldsymbol{p}_j \leq a_{max,j}\mathbf{1}_m \ (j = 1, 2, \cdots, N_{joint}) && (4.151)\end{aligned}$$

つまり，

$$
\begin{aligned}
& -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} \le \hat{\boldsymbol{D}}_1\boldsymbol{p} \le \hat{\boldsymbol{E}}\boldsymbol{v}_{max} && (4.152)\\
\Leftrightarrow \quad & \begin{pmatrix} \hat{\boldsymbol{D}}_1 \\ -\hat{\boldsymbol{D}}_1 \end{pmatrix} \boldsymbol{p} \ge \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} \\ -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} \end{pmatrix} && (4.153)\\
& -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} \le \hat{\boldsymbol{D}}_2\boldsymbol{p} \le \hat{\boldsymbol{E}}\boldsymbol{a}_{max} && (4.154)\\
\Leftrightarrow \quad & \begin{pmatrix} \hat{\boldsymbol{D}}_2 \\ -\hat{\boldsymbol{D}}_2 \end{pmatrix} \boldsymbol{p} \ge \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} \\ -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} \end{pmatrix} && (4.155)
\end{aligned}
$$

これは，逐次二次計画法の中で，次式の不等式制約となる．

$$
\begin{pmatrix} \hat{\boldsymbol{D}}_1 \\ -\hat{\boldsymbol{D}}_1 \end{pmatrix} \Delta\boldsymbol{p} \ge \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} - \hat{\boldsymbol{D}}_1\boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} + \hat{\boldsymbol{D}}_1\boldsymbol{p} \end{pmatrix} \tag{4.156}
$$

$$
\begin{pmatrix} \hat{\boldsymbol{D}}_2 \\ -\hat{\boldsymbol{D}}_2 \end{pmatrix} \Delta\boldsymbol{p} \ge \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} - \hat{\boldsymbol{D}}_2\boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} + \hat{\boldsymbol{D}}_2\boldsymbol{p} \end{pmatrix} \tag{4.157}
$$

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．

$$
\begin{aligned}
& t_s \le t_l \le t_f \quad (l = 1, 2, \cdots, N_{tm}) && (4.158)\\
\Leftrightarrow \quad & t_s\boldsymbol{1} \le \boldsymbol{t} \le t_f\boldsymbol{1} && (4.159)\\
\Leftrightarrow \quad & \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \boldsymbol{t} \ge \begin{pmatrix} t_s\boldsymbol{1} \\ -t_f\boldsymbol{1} \end{pmatrix} && (4.160)
\end{aligned}
$$

これは，逐次二次計画法の中で，次式の不等式制約となる．

$$
\begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{t} \ge \begin{pmatrix} t_s\boldsymbol{1} - \boldsymbol{t} \\ -t_f\boldsymbol{1} + \boldsymbol{t} \end{pmatrix} \tag{4.161}
$$

(4.162)

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．

$$
\begin{aligned}
& t_l \le t_{l+1} \quad (l = 1, 2, \cdots, N_{tm} - 1) && (4.163)\\
\Leftrightarrow \quad & -t_l + t_{l+1} \ge 0 \quad (l = 1, 2, \cdots, N_{tm} - 1) && (4.164)\\
\Leftrightarrow \quad & \boldsymbol{D}_{tm}\boldsymbol{t} \ge \boldsymbol{0} && (4.165)
\end{aligned}
$$

ただし，

$$
\boldsymbol{D}_{tm} = \begin{pmatrix} -1 & 1 & & & \boldsymbol{O} \\ & -1 & 1 & & \\ & & & \ddots & \\ \boldsymbol{O} & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(N_{tm}-1)\times N_{tm}} \tag{4.166}
$$

これは，逐次二次計画法の中で，次式の不等式制約となる．

$$
\boldsymbol{D}_{tm}\Delta\boldsymbol{t} \ge -\boldsymbol{D}_{tm}\boldsymbol{t} \tag{4.167}
$$

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．

$$F_{sqr,k}(\boldsymbol{p}) = \int_{t_s}^{t_f} \left\|\boldsymbol{\theta}^{(k)}(t)\right\|^2 dt \tag{4.168}$$

$$= \int_{t_s}^{t_f} \left\|\boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\boldsymbol{p}\right\|^2 dt \tag{4.169}$$

$$= \int_{t_s}^{t_f} \left(\boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\boldsymbol{p}\right)^T \left(\boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\boldsymbol{p}\right) dt \tag{4.170}$$

$$= \boldsymbol{p}^T \left\{\int_{t_s}^{t_f} \left(\boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\right)^T \boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k dt\right\}\boldsymbol{p} \tag{4.171}$$

$$= \boldsymbol{p}^T\boldsymbol{H}_k\boldsymbol{p} \tag{4.172}$$

ただし，

$$\boldsymbol{H}_k = \int_{t_s}^{t_f} \left(\boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\right)^T \boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k dt \tag{4.173}$$

$$\boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k = \begin{pmatrix} \boldsymbol{b}_{n-k}^T(t) & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \boldsymbol{b}_{n-k}^T(t) \end{pmatrix} \begin{pmatrix} (\boldsymbol{D}^k)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & (\boldsymbol{D}^k)^T \end{pmatrix} \tag{4.174}$$

$$= \begin{pmatrix} \boldsymbol{b}_{n-k}^T(t)(\boldsymbol{D}^k)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \boldsymbol{b}_{n-k}^T(t)(\boldsymbol{D}^k)^T \end{pmatrix} \tag{4.175}$$

$$= \begin{pmatrix} \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T \end{pmatrix} \tag{4.176}$$

$$\left(\boldsymbol{B}_{n-k}(t)\hat{\boldsymbol{D}}_k\right)^T \boldsymbol{B}_{n-k}(t) = \begin{pmatrix} \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T \end{pmatrix}^T \begin{pmatrix} \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T \end{pmatrix} \tag{4.177}$$

$$= \begin{pmatrix} \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)\left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)\left(\boldsymbol{D}^k\boldsymbol{b}_{n-k}(t)\right)^T \end{pmatrix} \tag{4.178}$$

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．

$$F_{duration}(\boldsymbol{t}) = |t_1 - t_{N_{tm}}|^2 \tag{4.179}$$

$$= \boldsymbol{t}^T \begin{pmatrix} 1 & & -1 \\ & & \\ -1 & & 1 \end{pmatrix} \boldsymbol{t} \tag{4.180}$$

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 $\boldsymbol{t}$ 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 $\boldsymbol{p}$)
(_timing-vector $\boldsymbol{t}$)
(_num-kin $N_{kin} := |\mathcal{T}^{kin\text{-}trg}| = |\mathcal{T}^{kin\text{-}att}|$)
(_num-joint $N_{joint} := |\mathcal{J}|$)
(_num-control-point N_{ctrl})
(_num-timing N_{tm})
(_bspline-order B-spline order, n)
(_dim-control-vector $dim(\boldsymbol{p})$)
(_dim-timing-vector $dim(\boldsymbol{t})$)
(_dim-config $dim(\boldsymbol{q})$)
(_dim-task $dim(\boldsymbol{e})$)
(_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 $\{t_1^{kin\text{-}tm}, t_2^{kin\text{-}tm}, \cdots, t_{N_{kin}}^{kin\text{-}tm}\}$)
(_kin-variable-timing-list list of bool. t for variable timing.)
(_kin-target-coords-list $\mathcal{T}^{kin\text{-}trg}$)
(_kin-attention-coords-list $\mathcal{T}^{kin\text{-}att}$)
(_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 $\boldsymbol{\theta}$ [deg] [mm])
(_collision-pair-list list of bodyset-link or body pair)
(_keep-timing-order? whether to keep order of timing $\boldsymbol{t}$ or not)
(_bspline-matrix buffer for $\boldsymbol{B}_n(t)$)

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

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 $\boldsymbol{q}$ とタスク関数 $\boldsymbol{e}(\boldsymbol{q})$ のクラス.

コンフィギュレーション $\boldsymbol{q}$ の取得・更新，タスク関数 $\boldsymbol{e}(\boldsymbol{q})$ の取得，タスク関数のヤコビ行列 $\frac{\partial \boldsymbol{e}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ の取得，コンフィギュレーションの等式・不等式制約 $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{C}, \boldsymbol{d}$ 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 $\mathcal{T}^{kin\text{-}trg}$ 幾何到達目標位置姿勢リスト

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

  kin-time-list $\{t_1^{kin\text{-}tm}, t_2^{kin\text{-}tm}, \cdots, t_{N_{kin}}^{kin\text{-}tm}\}$ 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 $\boldsymbol{t}$ の次元 N_{tm} となる.

コンフィギュレーション $\boldsymbol{q}$ は以下から構成される.

$$\boldsymbol{q} := \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{t} \end{pmatrix} \tag{4.181}$$

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

$\boldsymbol{t} \in \mathbb{R}^{N_{tm}}$ 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 $\boldsymbol{e}(\boldsymbol{q})$ は以下から構成される.

$$\boldsymbol{e}(\boldsymbol{q}) := \begin{pmatrix} \boldsymbol{e}^{kin}(\boldsymbol{q}) \\ \boldsymbol{e}^{stat}(\boldsymbol{q}) \end{pmatrix} \in \mathbb{R}^{6N_{kin}+4N_{joint}} \tag{4.182}$$

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

$\boldsymbol{e}^{stat}(\boldsymbol{q}) \in \mathbb{R}^{4N_{joint}}$ 初期，終端時刻静止拘束 [rad][rad/s][rad/s^2][m][m/s][m/s^2]

:init *&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 $N_{kin} := |\mathcal{T}^{kin\text{-}trg}| = |\mathcal{T}^{kin\text{-}att}|$

: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 $dim(\boldsymbol{q}) := dim(\boldsymbol{p}) + dim(\boldsymbol{t}) = N_{ctrl}N_{joint} + N_{tm}$

:dim-task [method]

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

:control-vector [method]

return control vector $\boldsymbol{p}$

:timing-vector [method]

return timing vector $\boldsymbol{t}$

:config-vector [method]

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

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

Set $\boldsymbol{p}$.

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

Set $\boldsymbol{t}$.

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

Set $\boldsymbol{q}$.

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

$$\boldsymbol{b}_n(t) := \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{N_{ctrl}-1,n}(t) \end{pmatrix} \in \mathbb{R}^{N_{ctrl}} \tag{4.183}$$

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

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

$$\boldsymbol{B}_n(t) := \begin{pmatrix} \boldsymbol{b}_n^T(t) & & & \boldsymbol{O} \\ & \boldsymbol{b}_n^T(t) & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{b}_n^T(t) \end{pmatrix} \in \mathbb{R}^{N_{joint} \times N_{ctrl} N_{joint}} \tag{4.184}$$

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

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

$$\boldsymbol{D} := \frac{1}{h} \begin{pmatrix} 1 & -1 & & & \boldsymbol{O} \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -1 \\ \boldsymbol{O} & & & & 1 \end{pmatrix} \in \mathbb{R}^{N_{ctrl} \times N_{ctrl}} \tag{4.185}$$

return $\boldsymbol{D}^k$

:extended-differential-matrix *&key (diff-order 1)* [method]

$$\hat{\boldsymbol{D}}_k \quad := \quad \begin{pmatrix} (\boldsymbol{D}^k)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & (\boldsymbol{D}^k)^T \end{pmatrix} \in \mathbb{R}^{N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}} \tag{4.186}$$

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

:bspline-differential-matrix *tm &key (diff-order 1)* [method]

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

:control-matrix [method]

$$\boldsymbol{P} := \begin{pmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_{n_{joint}}^T \end{pmatrix} \in \mathbb{R}^{N_{joint} \times N_{ctrl}} \tag{4.187}$$

return $\boldsymbol{P}$

:theta *tm* [method]

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

:theta-dot *tm &key (diff-order 1)* [method]

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

:theta-dot-numerical *tm &key (diff-order 1)*
(delta-time 0.05) [method]

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

:apply-theta-to-robot *tm* [method]

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

:kin-target-coords-list [method]

$$T_m^{kin\text{-}trg} = \{\boldsymbol{p}_l^{kin\text{-}trg}, \boldsymbol{R}_l^{kin\text{-}trg}\} \quad (l = 1, 2, \cdots, N_{kin}) \tag{4.188}$$

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

:kin-attention-coords-list [method]

$$T_m^{kin\text{-}att} = \{\boldsymbol{p}_l^{kin\text{-}att}, \boldsymbol{R}_l^{kin\text{-}att}\} \quad (l = 1, 2, \cdots, N_{kin}) \tag{4.189}$$

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

:kin-start-time [method]

return $t_s^{kin} := t_1^{kin\text{-}tm}$

:kin-finish-time [method]

return $t_f^{kin} := t_{N_{kin}}^{kin\text{-}tm}$

:motion-duration [method]

return $(t_{N_{kin}}^{kin\text{-}tm} - t_1^{kin\text{-}tm})$

:kinematics-task-value *&key (update? t)* [method]

$$\begin{aligned}
\boldsymbol{e}^{kin}(\boldsymbol{q}) &= \boldsymbol{e}^{kin}(\boldsymbol{p}, \boldsymbol{t}) && (4.190)\\
&= \begin{pmatrix} \boldsymbol{e}_1^{kin}(\boldsymbol{p}, \boldsymbol{t}) \\ \boldsymbol{e}_2^{kin}(\boldsymbol{p}, \boldsymbol{t}) \\ \vdots \\ \boldsymbol{e}_{N_{kin}}^{kin}(\boldsymbol{p}, \boldsymbol{t}) \end{pmatrix} && (4.191)\\
\boldsymbol{e}_l^{kin}(\boldsymbol{p}, \boldsymbol{t}) &= \begin{pmatrix} \boldsymbol{p}_l^{kin\text{-}trg} - \boldsymbol{p}_l^{kin\text{-}att}(\boldsymbol{p}, \boldsymbol{t}) \\ \boldsymbol{a}\left(\boldsymbol{R}_l^{kin\text{-}trg} \boldsymbol{R}_l^{kin\text{-}att}(\boldsymbol{p}, \boldsymbol{t})^T\right) \end{pmatrix} \in \mathbb{R}^6 \quad (l = 1, 2, \cdots, N_{kin}) && (4.192)
\end{aligned}$$

$\boldsymbol{a}(\boldsymbol{R})$ は姿勢行列 $\boldsymbol{R}$ の等価角軸ベクトルを表す.

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

:stationery-start-finish-task-value *&key (update? t)* [method]

$$\begin{aligned}
\boldsymbol{e}^{stat}(\boldsymbol{q}) &= \boldsymbol{e}^{stat}(\boldsymbol{p}, \boldsymbol{t}) && (4.193)\\
&= \begin{pmatrix} \boldsymbol{e}_{sv}^{stat}(\boldsymbol{p}, \boldsymbol{t}) \\ \boldsymbol{e}_{fv}^{stat}(\boldsymbol{p}, \boldsymbol{t}) \\ \boldsymbol{e}_{sa}^{stat}(\boldsymbol{p}, \boldsymbol{t}) \\ \boldsymbol{e}_{fa}^{stat}(\boldsymbol{p}, \boldsymbol{t}) \end{pmatrix} && (4.194)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{e}_{sv}^{stat}(\boldsymbol{p}, \boldsymbol{t}) &:= \dot{\boldsymbol{\theta}}(t_s^{kin}) && (4.195)\\
\boldsymbol{e}_{fv}^{stat}(\boldsymbol{p}, \boldsymbol{t}) &:= \dot{\boldsymbol{\theta}}(t_f^{kin}) && (4.196)\\
\boldsymbol{e}_{sa}^{stat}(\boldsymbol{p}, \boldsymbol{t}) &:= \ddot{\boldsymbol{\theta}}(t_s^{kin}) && (4.197)\\
\boldsymbol{e}_{fa}^{stat}(\boldsymbol{p}, \boldsymbol{t}) &:= \ddot{\boldsymbol{\theta}}(t_f^{kin}) && (4.198)
\end{aligned}$$

return $\boldsymbol{e}^{stat}(\boldsymbol{q}) \in \mathbb{R}^{4N_{joint}}$

:task-value *&key (update? t)* [method]

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

:kinematics-task-jacobian-with-control-vector [method]

式 (4.110) より，タスク関数 $\boldsymbol{e}^{kin}$ を制御点 $\boldsymbol{p}$ で微分したヤコビ行列は次式で得られる.

$$\begin{aligned}
\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{p}} &= \begin{pmatrix} \frac{\partial \boldsymbol{e}_1^{kin}}{\partial \boldsymbol{p}} \\ \frac{\partial \boldsymbol{e}_2^{kin}}{\partial \boldsymbol{p}} \\ \vdots \\ \frac{\partial \boldsymbol{e}_{N_{kin}}^{kin}}{\partial \boldsymbol{p}} \end{pmatrix} && (4.199)\\
\frac{\partial \boldsymbol{e}_l^{kin}}{\partial \boldsymbol{p}} &= -\boldsymbol{J}^{kin\text{-}att}(\boldsymbol{\theta}(t_l^{kin\text{-}tm}))\boldsymbol{B}_n(t_l^{kin\text{-}tm}) \quad (l = 1, 2, \cdots, N_{kin}) && (4.200)
\end{aligned}$$

return $\dfrac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{p}} \in \mathbb{R}^{6N_{kin} \times N_{ctrl}N_{joint}}$

:kinematics-task-jacobian-with-timing-vector [method]

式 (4.117) より，タスク関数 $\boldsymbol{e}^{kin}$ をタイミング $\boldsymbol{t}$ で微分したヤコビ行列は次式で得られる．

$$\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{t}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}_1}{\partial \boldsymbol{t}} \\ \frac{\partial \boldsymbol{e}^{kin}_2}{\partial \boldsymbol{t}} \\ \vdots \\ \frac{\partial \boldsymbol{e}^{kin}_{N_{kin}}}{\partial \boldsymbol{t}} \end{pmatrix} \tag{4.201}$$

$\frac{\partial \boldsymbol{e}^{kin}_l}{\partial \boldsymbol{t}}$ の i 番目の列ベクトル $\left[\frac{\partial \boldsymbol{e}^{kin}_l}{\partial \boldsymbol{t}}\right]_i \in \mathbb{R}^6$ は次式で表される ($i = 1, 2, \cdots, N_{tm}$).

$$\left[\frac{\partial \boldsymbol{e}^{kin}_l}{\partial \boldsymbol{t}}\right]_i = \begin{cases} -\boldsymbol{J}^{kin\text{-}att}(\boldsymbol{\theta}(t^{kin\text{-}tm}_l))\boldsymbol{P}\boldsymbol{D}\boldsymbol{b}_{n-1}(t^{kin\text{-}tm}_l) & t^{kin\text{-}tm}_l \text{ and } t_i \text{ is identical} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{4.202}$$

return $\dfrac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{t}} \in \mathbb{R}^{6N_{kin} \times N_{tm}}$

:stationery-start-finish-task-jacobian-with-control-vector [method]

式 (4.130) , 式 (4.131) , 式 (4.132) , 式 (4.133) より，タスク関数 $\boldsymbol{e}^{stat}$ を制御点 $\boldsymbol{p}$ で微分したヤコビ行列は次式で得られる．

$$\frac{\partial \boldsymbol{e}^{stat}}{\partial \boldsymbol{p}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{stat}_{sv}}{\partial \boldsymbol{p}} \\ \frac{\partial \boldsymbol{e}^{stat}_{fv}}{\partial \boldsymbol{p}} \\ \frac{\partial \boldsymbol{e}^{stat}_{sa}}{\partial \boldsymbol{p}} \\ \frac{\partial \boldsymbol{e}^{stat}_{fa}}{\partial \boldsymbol{p}} \end{pmatrix} \tag{4.203}$$

$$\frac{\partial \boldsymbol{e}^{stat}_{sv}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{p}} = \boldsymbol{B}_{n-1}(t^{kin}_s)\hat{\boldsymbol{D}}_1 \tag{4.204}$$

$$\frac{\partial \boldsymbol{e}^{stat}_{fv}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{p}} = \boldsymbol{B}_{n-1}(t^{kin}_f)\hat{\boldsymbol{D}}_1 \tag{4.205}$$

$$\frac{\partial \boldsymbol{e}^{stat}_{sa}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{p}} = \boldsymbol{B}_{n-2}(t^{kin}_s)\hat{\boldsymbol{D}}_2 \tag{4.206}$$

$$\frac{\partial \boldsymbol{e}^{stat}_{fa}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{p}} = \boldsymbol{B}_{n-2}(t^{kin}_f)\hat{\boldsymbol{D}}_2 \tag{4.207}$$

return $\dfrac{\partial \boldsymbol{e}^{stat}}{\partial \boldsymbol{p}} \in \mathbb{R}^{4N_{joint} \times N_{ctrl}N_{joint}}$

:stationery-start-finish-task-jacobian-with-timing-vector [method]

式 (4.134) , 式 (4.135) , 式 (4.136) , 式 (4.137) より，タスク関数 $\boldsymbol{e}^{stat}$ をタイミング $\boldsymbol{t}$ で微分したヤコビ行列は次式で得られる．

$$\frac{\partial \boldsymbol{e}^{stat}}{\partial \boldsymbol{t}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{stat}_{sv}}{\partial \boldsymbol{t}} \\ \frac{\partial \boldsymbol{e}^{stat}_{fv}}{\partial \boldsymbol{t}} \\ \frac{\partial \boldsymbol{e}^{stat}_{sa}}{\partial \boldsymbol{t}} \\ \frac{\partial \boldsymbol{e}^{stat}_{fa}}{\partial \boldsymbol{t}} \end{pmatrix} \tag{4.208}$$

$\frac{\partial \boldsymbol{e}_x^{stat}}{\partial \boldsymbol{t}}$ の i 番目の列ベクトル $\left[\frac{\partial \boldsymbol{e}_x^{stat}}{\partial \boldsymbol{t}}\right]_i \in \mathbb{R}^{N_{joint}}$ は次式で表される ($x \in \{sv, fv, sa, fa\}, i = 1, 2, \cdots, N_{tm}$).

$$\left[\frac{\partial \boldsymbol{e}_{sv}^{stat}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{t}}\right]_i = \begin{cases} \boldsymbol{P}\boldsymbol{D}^2\boldsymbol{b}_{n-2}(t_s^{kin}) & t_s^{kin} \text{ and } t_i \text{ is identical} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{4.209}$$

$$\left[\frac{\partial \boldsymbol{e}_{fv}^{stat}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{t}}\right]_i = \begin{cases} \boldsymbol{P}\boldsymbol{D}^2\boldsymbol{b}_{n-2}(t_f^{kin}) & t_f^{kin} \text{ and } t_i \text{ is identical} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{4.210}$$

$$\left[\frac{\partial \boldsymbol{e}_{sa}^{stat}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{t}}\right]_i = \begin{cases} \boldsymbol{P}\boldsymbol{D}^3\boldsymbol{b}_{n-3}(t_s^{kin}) & t_s^{kin} \text{ and } t_i \text{ is identical} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{4.211}$$

$$\left[\frac{\partial \boldsymbol{e}_{fa}^{stat}(\boldsymbol{p}, \boldsymbol{t})}{\partial \boldsymbol{t}}\right]_i = \begin{cases} \boldsymbol{P}\boldsymbol{D}^3\boldsymbol{b}_{n-3}(t_f^{kin}) & t_f^{kin} \text{ and } t_i \text{ is identical} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{4.212}$$

return $\frac{\partial \boldsymbol{e}^{stat}}{\partial \boldsymbol{t}} \in \mathbb{R}^{4N_{joint} \times N_{tm}}$

:task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} = \begin{array}{c} \\ 6N_{kin} \\ 4N_{joint} \end{array} \begin{array}{c} \begin{array}{cc} N_{ctrl}N_{joint} & N_{tm} \end{array} \\ \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{t}} \\ k_{stat}\frac{\partial \boldsymbol{e}^{stat}}{\partial \boldsymbol{p}} & k_{stat}\frac{\partial \boldsymbol{e}^{stat}}{\partial \boldsymbol{t}} \end{pmatrix} \end{array} \tag{4.213}$$

return $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} = \mathbb{R}^{(6N_{kin}+4N_{joint}) \times (N_{ctrl}N_{joint}+N_{tm})}$

:theta-max-vector *&key (update? nil)* [method]

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

:theta-min-vector *&key (update? nil)* [method]

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

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

式 (4.144) より，関節角度上下限制約は次式で表される．

$$\hat{\boldsymbol{E}}\boldsymbol{\theta}_{min} \leq \boldsymbol{p} + \Delta\boldsymbol{p} \leq \hat{\boldsymbol{E}}\boldsymbol{\theta}_{max} \tag{4.214}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{p} \geq \begin{pmatrix} \hat{\boldsymbol{E}}\boldsymbol{\theta}_{min} - \boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{\theta}_{max} + \boldsymbol{p} \end{pmatrix} \tag{4.215}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{\theta}}\Delta\boldsymbol{p} \geq \boldsymbol{d}_{\boldsymbol{\theta}} \tag{4.216}$$

ただし，

$$\hat{\boldsymbol{E}} := \begin{pmatrix} \boldsymbol{1}_{N_{ctrl}} & & & \boldsymbol{0}_{N_{ctrl}} \\ & \boldsymbol{1}_{N_{ctrl}} & & \\ & & \ddots & \\ \boldsymbol{0}_{N_{ctrl}} & & & \boldsymbol{1}_{N_{ctrl}} \end{pmatrix} \in \mathbb{R}^{N_{ctrl}N_{joint} \times N_{joint}} \tag{4.217}$$

$\boldsymbol{1}_{N_{ctrl}} \in \mathbb{R}^{N_{ctrl}}$ は全要素が 1 の N_{ctrl} 次元ベクトルである．

return $\boldsymbol{C}_{\boldsymbol{\theta}} := \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}}$

:theta-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{\boldsymbol{\theta}} := \begin{pmatrix} \hat{\boldsymbol{E}}\boldsymbol{\theta}_{min} - \boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{\theta}_{max} + \boldsymbol{p} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint}}$

:velocity-max-vector *&key (update? nil)* [method]

return $\boldsymbol{v}_{max} \in \mathbb{R}^{N_{joint}}$

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

式 (4.152) より，関節速度上下限制約は次式で表される．

$$
\begin{aligned}
& -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} \leq \hat{\boldsymbol{D}}_1(\boldsymbol{p} + \Delta\boldsymbol{p}) \leq \hat{\boldsymbol{E}}\boldsymbol{v}_{max} && (4.218)\\
\Leftrightarrow\quad & \begin{pmatrix} \hat{\boldsymbol{D}}_1 \\ -\hat{\boldsymbol{D}}_1 \end{pmatrix} \Delta\boldsymbol{p} \geq \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} - \hat{\boldsymbol{D}}_1\boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} + \hat{\boldsymbol{D}}_1\boldsymbol{p} \end{pmatrix} && (4.219)\\
\Leftrightarrow\quad & \boldsymbol{C}_{\dot{\boldsymbol{\theta}}}\Delta\boldsymbol{p} \geq \boldsymbol{d}_{\dot{\boldsymbol{\theta}}} && (4.220)
\end{aligned}
$$

return $\boldsymbol{C}_{\dot{\boldsymbol{\theta}}} := \begin{pmatrix} \hat{\boldsymbol{D}}_1 \\ -\hat{\boldsymbol{D}}_1 \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}}$

:velocity-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{\dot{\boldsymbol{\theta}}} := \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} - \hat{\boldsymbol{D}}_1\boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{v}_{max} + \hat{\boldsymbol{D}}_1\boldsymbol{p} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint}}$

:acceleration-max-vector *&key (update? nil)* [method]

return $\boldsymbol{a}_{max} \in \mathbb{R}^{N_{joint}}$

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

式 (4.154) より，関節加速度上下限制約は次式で表される．

$$
\begin{aligned}
& -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} \leq \hat{\boldsymbol{D}}_2(\boldsymbol{p} + \Delta\boldsymbol{p}) \leq \hat{\boldsymbol{E}}\boldsymbol{a}_{max} && (4.221)\\
\Leftrightarrow\quad & \begin{pmatrix} \hat{\boldsymbol{D}}_2 \\ -\hat{\boldsymbol{D}}_2 \end{pmatrix} \Delta\boldsymbol{p} \geq \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} - \hat{\boldsymbol{D}}_2\boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} + \hat{\boldsymbol{D}}_2\boldsymbol{p} \end{pmatrix} && (4.222)\\
\Leftrightarrow\quad & \boldsymbol{C}_{\ddot{\boldsymbol{\theta}}}\Delta\boldsymbol{p} \geq \boldsymbol{d}_{\ddot{\boldsymbol{\theta}}} && (4.223)
\end{aligned}
$$

return $\boldsymbol{C}_{\ddot{\boldsymbol{\theta}}} := \begin{pmatrix} \hat{\boldsymbol{D}}_2 \\ -\hat{\boldsymbol{D}}_2 \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint} \times N_{ctrl}N_{joint}}$

:acceleration-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{\ddot{\boldsymbol{\theta}}} := \begin{pmatrix} -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} - \hat{\boldsymbol{D}}_2\boldsymbol{p} \\ -\hat{\boldsymbol{E}}\boldsymbol{a}_{max} + \hat{\boldsymbol{D}}_2\boldsymbol{p} \end{pmatrix} \in \mathbb{R}^{2N_{ctrl}N_{joint}}$

:control-vector-inequality-constraint-matrix *&key (update? nil)* [method]

$$
\begin{aligned}
& \begin{cases} \boldsymbol{C}_{\boldsymbol{\theta}}\Delta\boldsymbol{p} \geq \boldsymbol{d}_{\boldsymbol{\theta}} \\ \boldsymbol{C}_{\dot{\boldsymbol{\theta}}}\Delta\boldsymbol{p} \geq \boldsymbol{d}_{\dot{\boldsymbol{\theta}}} \\ \boldsymbol{C}_{\ddot{\boldsymbol{\theta}}}\Delta\boldsymbol{p} \geq \boldsymbol{d}_{\ddot{\boldsymbol{\theta}}} \end{cases} && (4.224)\\
\Leftrightarrow\quad & \begin{pmatrix} \boldsymbol{C}_{\boldsymbol{\theta}} \\ \boldsymbol{C}_{\dot{\boldsymbol{\theta}}} \\ \boldsymbol{C}_{\ddot{\boldsymbol{\theta}}} \end{pmatrix} \Delta\boldsymbol{p} \geq \begin{pmatrix} \boldsymbol{d}_{\boldsymbol{\theta}} \\ \boldsymbol{d}_{\dot{\boldsymbol{\theta}}} \\ \boldsymbol{d}_{\ddot{\boldsymbol{\theta}}} \end{pmatrix} && (4.225)\\
\Leftrightarrow\quad & \boldsymbol{C}_{\boldsymbol{p}}\Delta\boldsymbol{p} \geq \boldsymbol{d}_{\boldsymbol{p}} && (4.226)
\end{aligned}
$$

return $\boldsymbol{C}_{\boldsymbol{p}} := \begin{pmatrix} \boldsymbol{C}_{\boldsymbol{\theta}} \\ \boldsymbol{C}_{\dot{\boldsymbol{\theta}}} \\ \boldsymbol{C}_{\ddot{\boldsymbol{\theta}}} \end{pmatrix} \in \mathbb{R}^{N_{p\text{-}ineq} \times dim(\boldsymbol{p})}$

:control-vector-inequality-constraint-vector *&key (update? t)* [method] return $\boldsymbol{d_p} := \begin{pmatrix} \boldsymbol{d_\theta} \\ \boldsymbol{d_{\dot{\theta}}} \\ \boldsymbol{d_{\ddot{\theta}}} \end{pmatrix} \in \mathbb{R}^{N_{p\text{-}ineq}}$

:timing-vector-inequality-constraint-matrix *&key (update? nil)* [method]

式 (4.159) 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．

$$t_s\boldsymbol{1} \leq \boldsymbol{t} + \Delta\boldsymbol{t} \leq t_f\boldsymbol{1} \tag{4.227}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{t} \geq \begin{pmatrix} t_s\boldsymbol{1} - \boldsymbol{t} \\ -t_f\boldsymbol{1} + \boldsymbol{t} \end{pmatrix} \tag{4.228}$$

また，式 (4.165) 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．

$$\boldsymbol{D}_{tm}(\boldsymbol{t} + \Delta\boldsymbol{t}) \geq \boldsymbol{0} \tag{4.229}$$

$$\Leftrightarrow \quad \boldsymbol{D}_{tm}\Delta\boldsymbol{t} \geq -\boldsymbol{D}_{tm}\boldsymbol{t} \tag{4.230}$$

ただし，

$$\boldsymbol{D}_{tm} = \begin{pmatrix} -1 & 1 & & & & \boldsymbol{O} \\ & -1 & 1 & & & \\ & & & \ddots & & \\ \boldsymbol{O} & & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(N_{tm}-1)\times N_{tm}} \tag{4.231}$$

これらを合わせると，

$$\begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \\ \boldsymbol{D}_{tm} \end{pmatrix} \Delta\boldsymbol{t} \geq \begin{pmatrix} t_s\boldsymbol{1} - \boldsymbol{t} \\ -t_f\boldsymbol{1} + \boldsymbol{t} \\ -\boldsymbol{D}_{tm}\boldsymbol{t} \end{pmatrix} \Leftrightarrow \quad \boldsymbol{C_t}\Delta\boldsymbol{p} \geq \boldsymbol{d_t} \tag{4.232}$$

return $\boldsymbol{C_t} := \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \\ \boldsymbol{D}_{tm} \end{pmatrix} \in \mathbb{R}^{(3N_{tm}-1)\times dim(\boldsymbol{t})}$

:timing-vector-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d_t} := \begin{pmatrix} t_s\boldsymbol{1} - \boldsymbol{t} \\ -t_f\boldsymbol{1} + \boldsymbol{t} \\ -\boldsymbol{D}_{tm}\boldsymbol{t} \end{pmatrix} \in \mathbb{R}^{(3N_{tm}-1)}$

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

$$\begin{cases} \boldsymbol{C_p}\Delta\boldsymbol{p} \geq \boldsymbol{d_p} \\ \boldsymbol{C_t}\Delta\boldsymbol{t} \geq \boldsymbol{d_t} \end{cases} \tag{4.233}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C_p} & \\ & \boldsymbol{C_t} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{p} \\ \Delta\boldsymbol{t} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d_p} \\ \boldsymbol{d_t} \end{pmatrix} \tag{4.234}$$

$$\Leftrightarrow \quad \boldsymbol{C}\Delta\boldsymbol{q} \geq \boldsymbol{d} \tag{4.235}$$

return $\boldsymbol{C} := \begin{pmatrix} \boldsymbol{C_p} & \\ & \boldsymbol{C_t} \end{pmatrix} \in \mathbb{R}^{N_{ineq}\times dim(\boldsymbol{q})}$

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

return $\boldsymbol{d} := \begin{pmatrix} \boldsymbol{d_p} \\ \boldsymbol{d_t} \end{pmatrix} \in \mathbb{R}^{N_{ineq}}$

:config-equality-constraint-matrix *&key (update? nil)* [method]

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

:config-equality-constraint-vector *&key (update? t)* [method]

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

:square-integration-regular-matrix *&key (diff-order 1)* *(delta-time (/ (- _finish-time _start-time) 100.0))* [method]

式 (4.172) より，関節角微分の二乗積分は次式で得られる．

$$F_{sqr,k}(\boldsymbol{p}) = \int_{t_s}^{t_f} \left\| \boldsymbol{\theta}^{(k)}(t) \right\|^2 dt \tag{4.236}$$

$$= \boldsymbol{p}^T \boldsymbol{H}_{sqr,k} \boldsymbol{p} \tag{4.237}$$

ただし，

$$\boldsymbol{H}_{sqr,k} = \int_{t_s}^{t_f} \left(\boldsymbol{B}_{n-k}(t) \hat{\boldsymbol{D}}_k \right)^T \boldsymbol{B}_{n-k}(t) \hat{\boldsymbol{D}}_k dt \tag{4.238}$$

$$= \int_{t_s}^{t_f} \begin{pmatrix} \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t)\right) \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t)\right)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t)\right) \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t)\right)^T \end{pmatrix} dt \tag{4.239}$$

これは二次形式の正則化項である．

return $\boldsymbol{H}_{sqr,k} \in \mathbb{R}^{dim(\boldsymbol{p}) \times dim(\boldsymbol{p})}$

:first-differential-square-integration-regular-matrix *&key (delta-time (/ (- _finish-time _start-time) 100.0))* [method]

return $\boldsymbol{H}_{sqr,1} \in \mathbb{R}^{dim(\boldsymbol{p}) \times dim(\boldsymbol{p})}$

:second-differential-square-integration-regular-matrix *&key (delta-time (/ (- _finish-time _start-time) 100.0))* [method]

return $\boldsymbol{H}_{sqr,2} \in \mathbb{R}^{dim(\boldsymbol{p}) \times dim(\boldsymbol{p})}$

:third-differential-square-integration-regular-matrix *&key (delta-time (/ (- _finish-time _start-time) 100.0))* [method]

return $\boldsymbol{H}_{sqr,3} \in \mathbb{R}^{dim(\boldsymbol{p}) \times dim(\boldsymbol{p})}$

:control-vector-regular-matrix [method]

$$\boldsymbol{W}_{reg,p} := \min(k_{max,p}, \|\boldsymbol{e}\|^2 + k_{off,p})\boldsymbol{I} + k_{sqr,1}\boldsymbol{H}_{sqr,1} + k_{sqr,2}\boldsymbol{H}_{sqr,2} + k_{sqr,3}\boldsymbol{H}_{sqr,3} \tag{4.240}$$

return $\boldsymbol{W}_{reg,p} \in \mathbb{R}^{dim(\boldsymbol{p}) \times dim(\boldsymbol{p})}$

:control-vector-regular-vector [method]

$$\boldsymbol{v}_{reg,p} := (k_{sqr,1}\boldsymbol{H}_{sqr,1} + k_{sqr,2}\boldsymbol{H}_{sqr,2} + k_{sqr,3}\boldsymbol{H}_{sqr,3})\boldsymbol{p} \tag{4.241}$$

return $\boldsymbol{v}_{reg,p} \in \mathbb{R}^{dim(\boldsymbol{p})}$

:motion-duration-regular-matrix [method]

式 (4.180) より，動作期間の二乗は次式で得られる．

$$\begin{aligned} F_{duration}(\boldsymbol{t}) &= |t_1 - t_{N_{tm}}|^2 && (4.242) \\ &= \boldsymbol{t}^T \begin{pmatrix} 1 & & -1 \\ & & \\ -1 & & 1 \end{pmatrix} \boldsymbol{t} && (4.243) \\ &= \boldsymbol{t}^T \boldsymbol{H}_{duration} \boldsymbol{t} && (4.244) \end{aligned}$$

これは二次形式の正則化項である．

return $\boldsymbol{H}_{duration} \in \mathbb{R}^{dim(\boldsymbol{t}) \times dim(\boldsymbol{t})}$

:timing-vector-regular-matrix [method]

$$\boldsymbol{W}_{reg,t} := \min(k_{max,t}, \|\boldsymbol{e}\|^2 + k_{off,t})\boldsymbol{I} + k_{duration}\boldsymbol{H}_{duration} \tag{4.245}$$

return $\boldsymbol{W}_{reg,t} \in \mathbb{R}^{dim(\boldsymbol{t}) \times dim(\boldsymbol{t})}$

:timing-vector-regular-vector [method]

$$\boldsymbol{v}_{reg,t} := k_{duration}\boldsymbol{H}_{duration}\boldsymbol{t} \tag{4.246}$$

return $\boldsymbol{v}_{reg,t} \in \mathbb{R}^{dim(\boldsymbol{t})}$

:regular-matrix [method]

$$\boldsymbol{W}_{reg} := \begin{pmatrix} \boldsymbol{W}_{reg,p} & \\ & \boldsymbol{W}_{reg,t} \end{pmatrix} \tag{4.247}$$

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

:regular-vector [method]

$$\boldsymbol{v}_{reg} := \begin{pmatrix} \boldsymbol{v}_{reg,p} \\ \boldsymbol{v}_{reg,t} \end{pmatrix} \tag{4.248}$$

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

: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 *&key* *(robot)* [method]
(delta-time (/ (- _finish-time _start-time) 100.0))
(only-motion-duration? t)
(loop? t)
(visualize-callback-func)

Play motion animation.

:plot-theta-graph *&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-configu

Plot graph.

:generate-angle-vector-sequence *&key* *(divide-num 100)* [method]
(start-time (send self :kin-start-time))
(finish-time (send self :kin-finish-time))
(delta-time (/ (float (- finish-time start-time)) divide-num))

Generate angle-vector-sequence.

get-bspline-knot *i n m x_min x_max h* [function]

$$\begin{aligned} t_i &= \frac{i-n}{m-n}(t_f - t_s) + t_s && (4.249) \\ &= hi + \frac{mt_s - nt_f}{m-n} && (4.250) \end{aligned}$$

return knot t_i for B-spline function

bspline-basis-func *x i n m x_min x_max &optional (n-orig n) (m-orig m)* [function]

$$\begin{aligned} b_{i,0}(t) &= \begin{cases} 1 & \text{if} \quad t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} && (4.251) \\ b_{i,n}(t) &= \frac{(t-t_i)b_{i,n-1}(t) + (t_{i+n+1}-t)b_{i+1,n-1}(t)}{nh} && (4.252) \end{aligned}$$

return B-spline function value $b_{i,n}(t)$.

4.3 B スプラインを用いた動的動作の生成

bspline-trajectory [class]

:super **propertied-object**
:slots (_start-time t_s)
(_finish-time t_f)
(_num-control-point N_{ctrl})
(_bspline-order B-spline order, n)
(_dim-instant-config $N_{\bar{q}}$)
(_dim-control-vector $dim(\boldsymbol{p}) := N_{ctrl}N_{\bar{q}}$)
(_control-vector $\boldsymbol{p}$)
(_zero-diff-stationery-start-finish-regular-scale $k_{stat,0}$)
(_first-diff-stationery-start-finish-regular-scale $k_{stat,1}$)
(_second-diff-stationery-start-finish-regular-scale $k_{stat,2}$)
(_diff-square-integration-regular-scale k_{sqr})
(_diff-mat buffer for $\boldsymbol{D}^k$)
(_diff-mat-list buffer for $\{\boldsymbol{D}^1, \boldsymbol{D}^2, \cdots, \boldsymbol{D}^K\}$)
(_extended-diff-mat-list buffer for $\{\hat{\boldsymbol{D}}_1, \hat{\boldsymbol{D}}_2, \cdots, \hat{\boldsymbol{D}}_K\}$)
(_ineq-const-matrix buffer for $\boldsymbol{C}_p$)
(_ineq-const-vector buffer for $\boldsymbol{d}_p$)

B スプラインを利用した軌道のクラス．

B スプラインベクトル・行列，制御点ベクトル・ベクトル，微分行列，瞬時コンフィギュレーションの取得や制御点ベクトルの更新のためのメソッドが定義されている．

B スプライン軌道を定めるために，初期化時に以下を与える

start-time t_s 初期時刻

finish-time t_f 終端時刻

num-control-point N_{ctrl} 制御点の個数

bspline-order n B スプラインのオーダー

dim-instant-config $N_{\bar{q}}$ 瞬時コンフィギュレーションの次元

ある時刻の瞬時コンフィギュレーション $\bar{\boldsymbol{q}}(t) \in \mathbb{R}^{N_{\bar{q}}}$ の j 番目の要素 $\bar{q}_j(t) \in \mathbb{R}$ を次式で表す．

$$\bar{q}_j(t) = \sum_{i=0}^{N_{ctrl}-1} p_{j,i} b_{i,n}(t) = \boldsymbol{p}_j^T \boldsymbol{b}_n(t) \quad (j = 1, 2, \cdots, N_{\bar{q}}) \tag{4.253}$$

ただし，

$$\boldsymbol{p}_j = \begin{pmatrix} p_{j,0} \\ p_{j,1} \\ \vdots \\ p_{j,N_{ctrl}-1} \end{pmatrix} \in \mathbb{R}^{N_{ctrl}} \quad (j = 1, 2, \cdots, N_{\bar{q}}) \tag{4.254}$$

$$\boldsymbol{b}_n(t) = \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{N_{ctrl}-1,n}(t) \end{pmatrix} \in \mathbb{R}^{N_{ctrl}} \tag{4.255}$$

$b_{i,n}(t)$ は B スプライン基底関数である．また，$p_{j,i}$ をそれぞれ制御点と呼ぶ．

したがって，$\bar{\boldsymbol{q}}(t)$ は次式で表される．

$$\bar{\boldsymbol{q}}(t) = \begin{pmatrix} \bar{q}_1(t) \\ \vdots \\ \bar{q}_{N_{\bar{q}}}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1^T \boldsymbol{b}_n(t) \\ \boldsymbol{p}_2^T \boldsymbol{b}_n(t) \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}}^T \boldsymbol{b}_n(t) \end{pmatrix} = \boldsymbol{P}\boldsymbol{b}_n(t) \tag{4.256}$$

ただし，

$$\boldsymbol{P} \quad = \quad \begin{pmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}}^T \end{pmatrix} \in \mathbb{R}^{N_{\bar{q}} \times N_{ctrl}} \tag{4.257}$$

また，$\bar{\boldsymbol{q}}(t)$ は，制御点を縦に並べたベクトルを分離して次式のようにも表される．

$$\bar{\boldsymbol{q}}(t) = \begin{pmatrix} \boldsymbol{b}_n^T(t)\boldsymbol{p}_1 \\ \boldsymbol{b}_n^T(t)\boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{b}_n^T(t)\boldsymbol{p}_{N_{\bar{q}}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_n^T(t) & & & \boldsymbol{O} \\ & \boldsymbol{b}_n^T(t) & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{b}_n^T(t) \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}} \end{pmatrix} = \boldsymbol{B}_n(t)\boldsymbol{p} \tag{4.258}$$

ただし，

$$\boldsymbol{B}_n(t) = \begin{pmatrix} \boldsymbol{b}_n^T(t) & & & \boldsymbol{O} \\ & \boldsymbol{b}_n^T(t) & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{b}_n^T(t) \end{pmatrix} \in \mathbb{R}^{N_{\bar{q}} \times N_{ctrl}N_{\bar{q}}}, \quad \boldsymbol{p} = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}} \end{pmatrix} \in \mathbb{R}^{N_{ctrl}N_{\bar{q}}} \tag{4.259}$$

B スプラインによる軌道表現の詳細については第**??**節参照．

:init *&key* *(name)* [method]
(start-time 0.0)
(finish-time 10.0)
(num-control-point 10)
(bspline-order 3)
(dim-instant-config 1)
(stationery-start-finish-regular-scale 1.0)
(zero-diff-stationery-start-finish-regular-scale 0.0)
(first-diff-stationery-start-finish-regular-scale stationery-start-finish-regular-scale)
(second-diff-stationery-start-finish-regular-scale stationery-start-finish-regular-scale)
(diff-square-integration-regular-scale 1.0)

Initialize instance

:start-time [method]

return t_s

:finish-time [method]

return t_s

:num-control-point [method]

return N_{ctrl}

:dim-instant-confing [method]

return $N_{\bar{q}}$

:dim-control-vector [method]

return $dim(\boldsymbol{p}) := N_{ctrl} N_{\bar{q}}$

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

$$\boldsymbol{b}_n(t) \quad := \quad \begin{pmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{N_{ctrl}-1,n}(t) \end{pmatrix} \in \mathbb{R}^{N_{ctrl}} \tag{4.260}$$

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

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

$$\boldsymbol{B}_n(t) := \begin{pmatrix} \boldsymbol{b}_n^T(t) & & & \boldsymbol{O} \\ & \boldsymbol{b}_n^T(t) & & \\ & & \ddots & \\ \boldsymbol{O} & & & \boldsymbol{b}_n^T(t) \end{pmatrix} \in \mathbb{R}^{N_{\bar{q}} \times N_{ctrl} N_{\bar{q}}} \tag{4.261}$$

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

:control-vector [method]

$$\boldsymbol{p} := \begin{pmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}} \end{pmatrix} \in \mathbb{R}^{N_{ctrl} N_{\bar{q}}} \tag{4.262}$$

return $\boldsymbol{p}$

:control-matrix [method]

$$\boldsymbol{P} := \begin{pmatrix} \boldsymbol{p}_1^T \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}}^T \end{pmatrix} \in \mathbb{R}^{N_{\bar{q}} \times N_{ctrl}} \tag{4.263}$$

return $\boldsymbol{P}$

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

$$\boldsymbol{D} \quad := \quad \frac{1}{h} \begin{pmatrix} 1 & -1 & & & \boldsymbol{O} \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -1 \\ \boldsymbol{O} & & & & 1 \end{pmatrix} \in \mathbb{R}^{N_{ctrl} \times N_{ctrl}} \tag{4.264}$$

return $\boldsymbol{D}^k$

:extended-differential-matrix *&key (diff-order 1)* [method]

$$\hat{\boldsymbol{D}}_k \quad := \quad \begin{pmatrix} (\boldsymbol{D}^k)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & (\boldsymbol{D}^k)^T \end{pmatrix} \in \mathbb{R}^{N_{ctrl}N_{\bar{q}} \times N_{ctrl}N_{\bar{q}}} \tag{4.265}$$

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

:instant-config *tm* [method]

return $\bar{\boldsymbol{q}}(t) = \begin{pmatrix} \boldsymbol{p}_1^T \boldsymbol{b}_n(t) \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}}^T \boldsymbol{b}_n(t) \end{pmatrix} = \boldsymbol{P}\boldsymbol{b}_n(t) = \boldsymbol{B}_n(t)\boldsymbol{p} \in \mathbb{R}^{N_{\bar{q}}}$

:instant-config-dot *tm &key (diff-order 1)* [method]

return $\bar{\boldsymbol{q}}^{(k)}(t) = \dfrac{d^{(k)}\bar{\boldsymbol{q}}(t)}{dt^{(k)}} = \boldsymbol{P}\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t)$

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

Set $\boldsymbol{p} \in \mathbb{R}^{N_{ctrl}N_{\bar{q}}}$.

:set-control-vector-from-instant-config *instant-config* [method]

Set $\boldsymbol{p} \in \mathbb{R}^{N_{ctrl}N_{\bar{q}}}$ from $\bar{\boldsymbol{q}} \in \mathbb{R}^{N_{\bar{q}}}$.

:convert-instant-inequality-constraint-matrix-for-control-vector *&key (instant-ineq-matrix) (update? nil)* [method]

$$\bar{\boldsymbol{q}}(t) = \begin{pmatrix} \bar{q}_1(t) \\ \vdots \\ \bar{q}_{N_{\bar{q}}}(t) \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{N_{ctrl}-1} p_{1,i} b_{i,n}(t) \\ \sum_{i=0}^{N_{ctrl}-1} p_{2,i} b_{i,n}(t) \\ \vdots \\ \sum_{i=0}^{N_{ctrl}-1} p_{N_{\bar{q}},i} b_{i,n}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1^T \boldsymbol{b}_n(t) \\ \boldsymbol{p}_2^T \boldsymbol{b}_n(t) \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}}^T \boldsymbol{b}_n(t) \end{pmatrix} = \boldsymbol{P}\boldsymbol{b}_n(t) \tag{4.266}$$

ただし,

$$\boldsymbol{P} \quad = \quad \begin{pmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}}^T \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{p}}_0 & \tilde{\boldsymbol{p}}_1 & \cdots & \tilde{\boldsymbol{p}}_{N_{ctrl}-1} \end{pmatrix} \tag{4.267}$$

$$\tilde{\boldsymbol{p}}_i \quad = \quad \begin{pmatrix} p_{1,i} \\ p_{2,i} \\ \vdots \\ p_{N_{\bar{q}},i} \end{pmatrix} \quad (i = 0, 1, \cdots, N_{ctrl} - 1) \tag{4.268}$$

ここで制御点 $\boldsymbol{p}$ が次式を満たすとする.

$$\boldsymbol{c}^T \tilde{\boldsymbol{p}}_i \geq d \quad (i = 0, 1, \cdots, N_{ctrl} - 1) \tag{4.269}$$

つまり,

$$\begin{pmatrix} c_1 & c_2 & \cdots & c_{N_{\bar{q}}} \end{pmatrix} \begin{pmatrix} p_{1,i} \\ p_{2,i} \\ \vdots \\ p_{N_{\bar{q}},i} \end{pmatrix} = \sum_{j=1}^{N_{\bar{q}}} c_j p_{j,i} \geq d \quad (i = 0, 1, \cdots, N_{ctrl} - 1) \tag{4.270}$$

このとき，

$$
\begin{aligned}
\boldsymbol{c}^T\bar{\boldsymbol{q}}(t) &= \begin{pmatrix} c_1 & c_2 & \cdots & c_{N_{\bar{q}}} \end{pmatrix} \begin{pmatrix} \sum_{i=0}^{N_{ctrl}-1} p_{1,i} b_{i,n}(t) \\ \sum_{i=0}^{N_{ctrl}-1} p_{2,i} b_{i,n}(t) \\ \vdots \\ \sum_{i=0}^{N_{ctrl}-1} p_{N_{\bar{q}},i} b_{i,n}(t) \end{pmatrix} && (4.271) \\
&= \sum_{j=1}^{N_{\bar{q}}} c_j \sum_{i=0}^{N_{ctrl}-1} p_{j,i} b_{i,n}(t) && (4.272) \\
&= \sum_{i=0}^{N_{ctrl}-1} \left(\sum_{j=1}^{N_{\bar{q}}} c_j p_{j,i} \right) b_{i,n}(t) && (4.273) \\
&\geq d \sum_{i=0}^{N_{ctrl}-1} b_{i,n}(t) && (4.274) \\
&= d && (4.275)
\end{aligned}
$$

したがって，

$$
\begin{aligned}
& \boldsymbol{C}_{\bar{q}} \bar{\boldsymbol{q}}(t) \geq \boldsymbol{d}_{\bar{q}} && (4.276) \\
\Leftrightarrow \quad & \boldsymbol{C}_{\bar{q}} \tilde{\boldsymbol{p}}_i \geq \boldsymbol{d}_{\bar{q}} \quad (i = 0, 1, \cdots, N_{ctrl} - 1) && (4.277) \\
\Leftrightarrow \quad & \begin{array}{c} \\ N_{ineq} \\ N_{ineq} \\ \\ N_{ineq} \end{array} \begin{array}{c} \begin{array}{cccc} N_{ctrl} & N_{ctrl} & & N_{ctrl} \end{array} \\ \begin{pmatrix} \boldsymbol{C}_{p,0,1} & \boldsymbol{C}_{p,0,2} & \cdots & \boldsymbol{C}_{p,0,N_{\bar{q}}} \\ \boldsymbol{C}_{p,1,1} & \boldsymbol{C}_{p,1,2} & \cdots & \boldsymbol{C}_{p,1,N_{\bar{q}}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{p,N_{ctrl}-1,1} & \boldsymbol{C}_{p,N_{ctrl}-1,2} & \cdots & \boldsymbol{C}_{p,N_{ctrl}-1,N_{\bar{q}}} \end{pmatrix} \end{array} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_{N_{\bar{q}}} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_{\bar{q}} \\ \boldsymbol{d}_{\bar{q}} \\ \vdots \\ \boldsymbol{d}_{\bar{q}} \end{pmatrix} && (4.278) \\
\Leftrightarrow \quad & \boldsymbol{C}_p \boldsymbol{p} \geq \boldsymbol{d}_p && (4.279)
\end{aligned}
$$

ただし，

$$
\begin{aligned}
\boldsymbol{C}_{\bar{q}} &= \begin{pmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_{N_{\bar{q}}} \end{pmatrix} \in \mathbb{R}^{N_{ineq} \times N_{\bar{q}}} && (4.280) \\
\boldsymbol{C}_{p,i,j} &= \begin{array}{c} \begin{array}{ccccccc} & & & i\text{番目} & & & \end{array} \\ \begin{pmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{c}_j & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{pmatrix} \end{array} \in \mathbb{R}^{N_{ineq} \times N_{ctrl}} && (4.281) \\
& \quad (i = 0, 1, \cdots, N_{ctrl} - 1, \ j = 1, 2, \cdots, N_{\bar{q}}) && (4.282)
\end{aligned}
$$

このメソッドは $\boldsymbol{C}_{\bar{q}} \in \mathbb{R}^{N_{ineq} \times N_{\bar{q}}}$ を受け取り，$\boldsymbol{C}_p \in \mathbb{R}^{N_{ctrl} N_{ineq} \times N_{ctrl} N_{\bar{q}}}$ を返す.

:convert-instant-inequality-constraint-vector-for-control-vector *&key (instant-ineq-vector) (update? nil)* [method]

このメソッドは $\boldsymbol{d}_{\bar{q}} \in \mathbb{R}^{N_{ineq}}$ を受け取り，$\boldsymbol{d}_p \in \mathbb{R}^{N_{ctrl} N_{ineq}}$ を返す.

:stationery-start-finish-regular-matrix *&key (start-time _start-time) (finish-time _finish-time) (update? nil)* [method]

$$
\begin{aligned}
\boldsymbol{W}_{stat} &= k_{stat,0} \boldsymbol{B}_n^T(t_s) \boldsymbol{B}_n(t_s) + k_{stat,0} \boldsymbol{B}_n^T(t_f) \boldsymbol{B}_n(t_f) && (4.283) \\
&+ k_{stat,1} (\boldsymbol{B}_{n-1}(t_s) \hat{\boldsymbol{D}}_1)^T (\boldsymbol{B}_{n-1}(t_s) \hat{\boldsymbol{D}}_1) + k_{stat,1} (\boldsymbol{B}_{n-1}(t_f) \hat{\boldsymbol{D}}_1)^T (\boldsymbol{B}_{n-1}(t_f) \hat{\boldsymbol{D}}_1) && (4.284) \\
&+ k_{stat,2} (\boldsymbol{B}_{n-2}(t_s) \hat{\boldsymbol{D}}_2)^T (\boldsymbol{B}_{n-2}(t_s) \hat{\boldsymbol{D}}_2) + k_{stat,2} (\boldsymbol{B}_{n-2}(t_f) \hat{\boldsymbol{D}}_2)^T (\boldsymbol{B}_{n-2}(t_f) \hat{\boldsymbol{D}}_2) && (4.285)
\end{aligned}
$$

return $\boldsymbol{W}_{stat} \in \mathbb{R}^{N_{ctrl} N_{\bar{q}} \times N_{ctrl} N_{\bar{q}}}$

:differential-square-integration-regular-matrix *&key (start-time _start-time) (finish-time _finish-time) (delta-time (/ (- finish-time start-time) 100.0)) (diff-order 1)* [method]

式 (4.172) より，コンフィギュレーション微分の二乗積分は次式で得られる．

$$
\begin{aligned}
F_{sqr,k}(\boldsymbol{p}) &= \int_{t_s}^{t_f} \left\| \bar{\boldsymbol{q}}^{(k)}(t) \right\|^2 dt && (4.286) \\
&= \boldsymbol{p}^T \boldsymbol{W}_{sqr,k} \boldsymbol{p} && (4.287)
\end{aligned}
$$

ただし，

$$
\begin{aligned}
\boldsymbol{W}_{sqr,k} &= \int_{t_s}^{t_f} \left(\boldsymbol{B}_{n-k}(t) \hat{\boldsymbol{D}}_k \right)^T \boldsymbol{B}_{n-k}(t) \hat{\boldsymbol{D}}_k dt && (4.288) \\
&= \int_{t_s}^{t_f} \begin{pmatrix} \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t) \right) \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t) \right)^T & & \boldsymbol{O} \\ & \ddots & \\ \boldsymbol{O} & & \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t) \right) \left(\boldsymbol{D}^k \boldsymbol{b}_{n-k}(t) \right)^T \end{pmatrix} dt && (4.289)
\end{aligned}
$$

return $k_{sqr} \boldsymbol{W}_{sqr,k} \in \mathbb{R}^{dim(\boldsymbol{p}) \times dim(\boldsymbol{p})}$

:dump-config-data *&key (start-time _start-time) (finish-time _finish-time) (delta-time (/ (- finish-time start-time) 100.0)) (data-filename (format nil /tmp/ a.dat (send self :name))) (diff-order 0)* [method]

bspline-dynamic-configuration-task [class]

:super **propertied-object**

:slots

(_robot-env robot-environment instance)
(_theta-bst bspline trajectory instance for $\boldsymbol{\theta}$)
(_cog-bst bspline trajectory instance for $\boldsymbol{c}$)
(_ang-moment-bst bspline trajectory instance for $\boldsymbol{L}$)
(_wrench-bst bspline trajectory instance for $\hat{\boldsymbol{w}}$)
(_torque-bst bspline trajectory instance for $\boldsymbol{\tau}$)
(_phi-vector $\boldsymbol{\phi}$)
(_num-kin $N_{kin} := |\mathcal{T}^{kin\text{-}trg}| = |\mathcal{T}^{kin\text{-}att}|$)
(_num-contact $N_{cnt} := |\mathcal{T}^{cnt\text{-}trg}| = |\mathcal{T}^{cnt\text{-}att}|$)
(_num-variant-joint $N_{var\text{-}joint} := |\mathcal{J}_{var}|$)
(_num-invariant-joint $N_{invar\text{-}joint} := |\mathcal{J}_{invar}|$)
(_num-drive-joint $N_{drive\text{-}joint} := |\mathcal{J}_{drive}|$)
(_num-posture-joint $N_{posture\text{-}joint} := |\mathcal{J}_{posture}|$)
(_num-collision N_{col} := number of collision check pairs)
(_dim-theta-control-vector $dim(\boldsymbol{p}_\theta) := N_{var\text{-}joint} N_{\theta\text{-}ctrl}$)
(_dim-cog-control-vector $dim(\boldsymbol{p}_c) := 3N_{c\text{-}ctrl}$)
(_dim-ang-moment-control-vector $dim(\boldsymbol{p}_L) := 3N_{L\text{-}ctrl}$)
(_dim-wrench-control-vector $dim(\boldsymbol{p}_{\hat{w}}) := 6N_{cnt} N_{\hat{w}\text{-}ctrl}$)
(_dim-torque-control-vector $dim(\boldsymbol{p}_\tau) := N_{drive\text{-}joint} N_{\tau\text{-}ctrl}$)
(_dim-phi $dim(\boldsymbol{\phi}) := N_{invar\text{-}joint}$)
(_dim-config $dim(\boldsymbol{q}) := dim(\boldsymbol{p}_\theta) + dim(\boldsymbol{p}_c) + dim(\boldsymbol{p}_L) + dim(\boldsymbol{p}_{\hat{w}}) + dim(\boldsymbol{p}_\tau) + dim(\boldsymbol{\phi})$)
(_dim-kin-task $dim(\boldsymbol{e}^{kin})$)
(_dim-eom-trans-task $dim(\boldsymbol{e}^{eom\text{-}trans})$)

(_dim-eom-rot-task $dim(\boldsymbol{e}^{eom\text{-}rot})$)
(_dim-cog-task $dim(\boldsymbol{e}^{cog})$)
(_dim-ang-moment-task $dim(\boldsymbol{e}^{ang\text{-}moment})$)
(_dim-torque-task $dim(\boldsymbol{e}^{trq})$)
(_dim-posture-task $dim(\boldsymbol{e}^{posture})$)
(_dim-task $dim(\boldsymbol{e})$)
(_kin-task-scale k_{kin})
(_kin-task-scale-mat-list-func function returning list of K_{kin})
(_eom-trans-task-scale $k_{eom\text{-}trans}$)
(_eom-rot-task-scale $k_{eom\text{-}rot}$)
(_cog-task-scale k_{cog})
(_ang-moment-task-scale $k_{ang\text{-}moment}$)
(_torque-task-scale k_{trq})
(_posture-task-scale $k_{posture}$)
(_torque-regular-scale k_{trq})
(_stationery-start-finish-regular-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 $k_{sqr,3}$)
(_norm-regular-scale-max k_{max})
(_norm-regular-scale-offset k_{off})
(_variant-joint-list $\mathcal{J}_{var}$)
(_invariant-joint-list $\mathcal{J}_{invar}$)
(_drive-joint-list $\mathcal{J}_{drive}$)
(_posture-joint-list $\mathcal{J}_{posture}$)
(_kin-task-time-list time list for kinematics task)
(_eom-task-time-list time list for eom task)
(_centroid-task-time-list time list for centroid task)
(_posture-task-time-list time list for posture task)
(_kin-target-coords-list-func function returning $\mathcal{T}^{kin\text{-}trg}$)
(_kin-attention-coords-list-func function returning $\mathcal{T}^{kin\text{-}att}$)
(_contact-target-coords-list-func function returning $\mathcal{T}^{cnt\text{-}trg}$)
(_contact-attention-coords-list-func function returning $\mathcal{T}^{cnt\text{-}att}$)
(_contact-constraint-list-func function returning list of contact-constraint)
(_posture-joint-angle-list $\bar{\boldsymbol{\theta}}^{trg}$)
(_variant-joint-angle-margin margin of $\boldsymbol{\theta}$ [deg] [mm])
(_invariant-joint-angle-margin margin of $\boldsymbol{\phi}$ [deg] [mm])
(_collision-pair-list list of bodyset-link or body pair)
(_collision-distance-margin-list list of collision distance margin)
(_task-jacobi buffer for $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}}$)
(_collision-theta-inequality-constraint-matrix buffer for $\boldsymbol{C}_{col,\theta}$)
(_collision-phi-inequality-constraint-matrix buffer for $\boldsymbol{C}_{col,\phi}$)
(_collision-inequality-constraint-vector buffer for $\boldsymbol{C}_{col}$)

B スプラインを利用した動的動作生成のためのコンフィギュレーション $\boldsymbol{q}$ とタスク関数 $\boldsymbol{e}(\boldsymbol{q})$ のクラス.

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

初期化

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)

- B スプライン軌道

  theta-bst 時変関節位置 $\boldsymbol{\theta}$ の B スプライン軌道のインスタンス

  cog-bst 重心位置 $\boldsymbol{c}$ の B スプライン軌道のインスタンス

  ang-moment-bst 角運動量 $\boldsymbol{L}$ の B スプライン軌道のインスタンス

  wrench-bst 接触レンチ $\hat{\boldsymbol{w}}$ の B スプライン軌道のインスタンス

  torque-bst 関節トルク $\boldsymbol{\tau}$ の B スプライン軌道のインスタンス

- タスク関数のサンプリング時刻

  kin-task-time-list 幾何到達拘束 $\boldsymbol{e}^{kin}$ の時刻のリスト

  eom-task-time-list 並進運動方程式 $\boldsymbol{e}^{eom\text{-}trans}$，回転運動方程式 $\boldsymbol{e}^{eom\text{-}rot}$ の時刻リスト

  centroid-task-time-list 重心位置 $\boldsymbol{e}^{cog}$，角運動量 $\boldsymbol{e}^{ang\text{-}moment}$ の時刻リスト

  posture-task-time-list 関節角目標 $\boldsymbol{e}^{posture}$ の時刻リスト

- 幾何拘束

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

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

- 接触拘束

  contact-target-coords-list-func 接触目標位置姿勢リスト $\mathcal{T}^{cnt\text{-}trg}$ を返す関数

  contact-attention-coords-list-func 接触着目位置姿勢リスト $\mathcal{T}^{cnt\text{-}att}$ を返す関数

  contact-constraint-list-func 接触レンチ制約リストを返す関数

- 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 $\bar{\boldsymbol{\theta}}^{trg}$ 着目関節の目標関節角

- 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)

- 目的関数の重み

  kin-task-scale k_{kin} 幾何到達拘束タスクの重み

  kin-task-scale-mat-list-func 幾何到達拘束タスクの重み行列 K_{kin} を返す関数

  eom-trans-task-scale $k_{eom\text{-}trans}$ 並進運動方程式タスクの重み

eom-rot-task-scale $k_{eom\text{-}rot}$ 回転運動方程式タスクの重み

cog-task-scale k_{cog} 重心位置タスクの重み

ang-moment-task-scale $k_{ang\text{-}moment}$ 角運動量タスクの重み

torque-task-scale k_{trq} 関節トルクの釣り合いタスクの重み

posture-task-scale $k_{posture}$ 目標関節角タスクの重み

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

stationery-start-finish-regular-scale k_{stat} 初期・終端静止正則化の重み

first-diff-square-integration-regular-scale $k_{sqr,1}$ 速度正則化の重み

second-diff-square-integration-regular-scale $k_{sqr,2}$ 加速度正則化の重み

third-diff-square-integration-regular-scale $k_{sqr,3}$ 躍度正則化の重み

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

コンフィギュレーション

動的動作は各瞬間において以下の瞬時状態 $\bar{\boldsymbol{q}}(t)$ を定めることで表現される．

$$\bar{\boldsymbol{q}}(t) := \begin{pmatrix} \boldsymbol{\theta}(t) \\ \boldsymbol{c}(t) \\ \boldsymbol{L}(t) \\ \hat{\boldsymbol{w}}(t) \\ \boldsymbol{\tau}(t) \\ \boldsymbol{\phi} \end{pmatrix} \tag{4.290}$$

$\boldsymbol{\theta} \in \mathbb{R}^{N_{var\text{-}joint}}$ 時変関節位置 [rad] [m]

$\boldsymbol{c} \in \mathbb{R}^3$ 重心位置 [m]

$\boldsymbol{L} \in \mathbb{R}^3$ 角運動量 [kgm^2/s]

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

$\boldsymbol{\tau} \in \mathbb{R}^{N_{drive\text{-}joint}}$ 関節トルク [Nm] [N]

$\boldsymbol{\phi} \in \mathbb{R}^{N_{invar\text{-}joint}}$ 時不変関節位置 [rad] [m]

$\hat{\boldsymbol{w}}$ 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．

$$\begin{aligned} \hat{\boldsymbol{w}} &= \begin{pmatrix} \boldsymbol{w}_1^T & \boldsymbol{w}_2^T & \cdots & \boldsymbol{w}_{N_{cnt}}^T \end{pmatrix}^T & (4.291) \\ &= \begin{pmatrix} \boldsymbol{f}_1^T & \boldsymbol{n}_1^T & \boldsymbol{f}_2^T & \boldsymbol{n}_2^T & \cdots & \boldsymbol{f}_{N_{cnt}}^T & \boldsymbol{n}_{N_{cnt}}^T \end{pmatrix}^T & (4.292) \end{aligned}$$

本クラスでは，瞬時状態 $\bar{\boldsymbol{q}}(t)$ の軌道を B スプラインで表現する．設計対称のコンフィギュレーション $\boldsymbol{q}$ は以下から構成される．

$$\boldsymbol{q} := \begin{pmatrix} \boldsymbol{p}_\theta \\ \boldsymbol{p}_c \\ \boldsymbol{p}_L \\ \boldsymbol{p}_{\hat{w}} \\ \boldsymbol{p}_\tau \\ \boldsymbol{\phi} \end{pmatrix} \tag{4.293}$$

$\boldsymbol{p}_\theta \in \mathbb{R}^{N_{var\text{-}joint}N_{\theta\text{-}ctrl}}$ 時変関節位置の制御点 [rad] [m]

$\boldsymbol{p}_c \in \mathbb{R}^{3N_{c\text{-}ctrl}}$ 重心位置の制御点 [m]

$\boldsymbol{p}_L \in \mathbb{R}^{3N_{L\text{-}ctrl}}$ 角運動量の制御点 [kgm^2/s]

$\boldsymbol{p}_{\hat{w}} \in \mathbb{R}^{6N_{cnt}N_{\hat{w}\text{-}ctrl}}$ 接触レンチの制御点 [N] [Nm]

$\boldsymbol{p}_\tau \in \mathbb{R}^{N_{drive\text{-}joint}N_{\tau\text{-}ctrl}}$ 関節トルクの制御点 [Nm] [N]

$\boldsymbol{\phi} \in \mathbb{R}^{N_{invar\text{-}joint}}$ 時不変関節位置 [rad] [m]

制御点とは，B スプライン基底関数の重み係数を意味する．

タスク関数

瞬時状態 $\bar{\boldsymbol{q}}(t)$ に関するタスク関数 $\bar{\boldsymbol{e}}(\bar{\boldsymbol{q}}(t))$ は以下から構成される．

$$\bar{\boldsymbol{e}}(\bar{\boldsymbol{q}}) = \begin{pmatrix} \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}, \hat{\boldsymbol{w}}) \\ \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{L}, \hat{\boldsymbol{w}}, \boldsymbol{\phi}) \\ \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{\phi}) \\ \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{\theta}, \boldsymbol{L}, \boldsymbol{\phi}) \\ \bar{\boldsymbol{e}}^{trq}(\boldsymbol{\theta}, \hat{\boldsymbol{w}}, \boldsymbol{\tau}, \boldsymbol{\phi}) \\ \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}) \end{pmatrix} \tag{4.294}$$

$\bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \in \mathbb{R}^{6\bar{N}_{kin}(t)}$ 幾何到達拘束 [rad] [m]

$$\bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \begin{pmatrix} \bar{\boldsymbol{e}}_1^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \bar{\boldsymbol{e}}_2^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \vdots \\ \bar{\boldsymbol{e}}_{\bar{N}_{kin}(t)}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \end{pmatrix} \tag{4.295}$$

$$\bar{\boldsymbol{e}}_m^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \begin{pmatrix} \boldsymbol{p}_m^{kin\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}_m^{kin\text{-}att}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \boldsymbol{a}\left(\boldsymbol{R}_m^{kin\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi})\boldsymbol{R}_m^{kin\text{-}att}(\boldsymbol{\theta}, \boldsymbol{\phi})^T\right) \end{pmatrix} \in \mathbb{R}^6 \quad (m = 1, 2, \cdots, \bar{N}_{kin}(t)) \tag{4.296}$$

$\boldsymbol{a}(\boldsymbol{R})$ は姿勢行列 $\boldsymbol{R}$ の等価角軸ベクトルを表す．

$\bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}, \hat{\boldsymbol{w}}) \in \mathbb{R}^3$ 並進運動方程式 [kg m/s^2]

$$\bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}, \hat{\boldsymbol{w}}) = m\ddot{\boldsymbol{c}} - \left\{ \sum_{m=1}^{N_{cnt}} \boldsymbol{f}_m - m\boldsymbol{g} + \sum_{m=1}^{N_{ex}} \boldsymbol{f}_m^{ex} \right\} \tag{4.297}$$

$\bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{L}, \hat{\boldsymbol{w}}, \boldsymbol{\phi}) \in \mathbb{R}^3$ 回転運動方程式 [kg m^2/s^2]

$$\begin{aligned} \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{L}, \hat{\boldsymbol{w}}, \boldsymbol{\phi}) &= \dot{\boldsymbol{L}} - \left(\sum_{m=1}^{N_{cnt}} \left\{ \left(\boldsymbol{p}_m^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c}\right) \times \boldsymbol{f}_m + \boldsymbol{n}_m \right\} \right. \\ &\quad \left. + \sum_{m=1}^{N_{ex}} \left\{ (\boldsymbol{p}_m^{ex}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c}) \times \boldsymbol{f}_m^{ex} + \boldsymbol{n}_m^{ex} \right\} \right) \end{aligned} \tag{4.298}$$

$\bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{\phi}) \in \mathbb{R}^3$ 重心位置 [m]

$$\bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{\phi}) = \boldsymbol{p}_G(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c} \tag{4.299}$$

$\bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{\theta}, \boldsymbol{L}, \boldsymbol{\phi}) \in \mathbb{R}^3$ 角運動量 [kg m^2/s]

$$\bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{\theta}, \boldsymbol{L}, \boldsymbol{\phi}) = \boldsymbol{L} - \left\{ \boldsymbol{A}_\theta(\boldsymbol{\theta}, \boldsymbol{\phi})\dot{\boldsymbol{\theta}} + \boldsymbol{A}_\phi(\boldsymbol{\theta}, \boldsymbol{\phi})\dot{\boldsymbol{\phi}} \right\} \tag{4.300}$$

$\bar{\boldsymbol{e}}^{trq}(\boldsymbol{\theta}, \hat{\boldsymbol{w}}, \boldsymbol{\tau}, \boldsymbol{\phi}) \in \mathbb{R}^{N_{drive\text{-}joint}}$ 関節トルクの釣り合い [rad] [m]

$$\begin{aligned} \bar{\boldsymbol{e}}^{trq}(\boldsymbol{\theta}, \hat{\boldsymbol{w}}, \boldsymbol{\tau}, \boldsymbol{\phi}) &= \sum_{m=1}^{N_{cnt}} \left\{ \left(\boldsymbol{p}_m^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c} \right) \times \boldsymbol{f}_m + \boldsymbol{n}_m \right\} \\ &\quad + \sum_{m=1}^{N_{ex}} \left\{ \left(\boldsymbol{p}_m^{ex}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c} \right) \times \boldsymbol{f}_m^{ex} + \boldsymbol{n}_m^{ex} \right\} \end{aligned} \tag{4.301}$$

$\bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}) \in \mathbb{R}^{N_{posture\text{-}joint}}$ 関節角目標 [rad] [m]

$$\bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}) = \left(\bar{\boldsymbol{\theta}}^{trg} - \bar{\boldsymbol{\theta}} \right) \tag{4.302}$$

瞬時状態 $\bar{\boldsymbol{q}}(t)$ の軌道を B スプラインで表現することで，設計対称のコンフィギュレーション $\boldsymbol{q}$ に関するタスク関数 $\boldsymbol{e}(\boldsymbol{q})$ は以下から構成される．

$$\boldsymbol{e}(\boldsymbol{q}) = \begin{pmatrix} \boldsymbol{e}^{kin}(\boldsymbol{p}_\theta, \boldsymbol{\phi}) \\ \boldsymbol{e}^{eom\text{-}trans}(\boldsymbol{p}_c, \boldsymbol{p}_{\hat{w}}) \\ \boldsymbol{e}^{eom\text{-}rot}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{p}_L, \boldsymbol{p}_{\hat{w}}, \boldsymbol{\phi}) \\ \boldsymbol{e}^{cog}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{\phi}) \\ \boldsymbol{e}^{ang\text{-}moment}(\boldsymbol{p}_\theta, \boldsymbol{p}_L, \boldsymbol{\phi}) \\ \boldsymbol{e}^{trq}(\boldsymbol{p}_\theta, \boldsymbol{p}_{\hat{w}}, \boldsymbol{p}_\tau, \boldsymbol{\phi}) \\ \boldsymbol{e}^{posture}(\boldsymbol{p}_\theta) \end{pmatrix} \tag{4.303}$$

ただし，

$$\boldsymbol{e}^*(\boldsymbol{q}) = \begin{pmatrix} \bar{\boldsymbol{e}}^*(\bar{\boldsymbol{q}}(t_1)) \\ \vdots \\ \bar{\boldsymbol{e}}^*(\bar{\boldsymbol{q}}(t_{N_{tm}})) \end{pmatrix} \in \mathbb{R}^{N_{tm}\, dim(\bar{\boldsymbol{e}}^*(\bar{\boldsymbol{q}})(t))} \tag{4.304}$$

:init *&key* *(name)* [method]
(robot-env)
(variant-joint-list (send robot-env :variant-joint-list))
(invariant-joint-list (send robot-env :invariant-joint-list))
(drive-joint-list (send robot-env :drive-joint-list))
(posture-joint-list)
(kin-task-time-list)
(eom-task-time-list)
(centroid-task-time-list)
(posture-task-time-list)
(theta-bst)
(cog-bst)
(ang-moment-bst)
(wrench-bst)
(torque-bst)
(kin-target-coords-list-func)
(kin-attention-coords-list-func)

(contact-target-coords-list-func)
(contact-attention-coords-list-func)
(contact-constraint-list-func)
(posture-joint-angle-list)
(variant-joint-angle-margin 3.0)
(invariant-joint-angle-margin 3.0)
(collision-pair-list)
(collision-distance-margin 0.01)
(collision-distance-margin-list)
(kin-task-scale 1.0)
(kin-task-scale-mat-list-func)
(eom-trans-task-scale 1.0)
(eom-rot-task-scale 1.0)
(cog-task-scale 1.0)
(ang-moment-task-scale 1.0)
(torque-task-scale 1.0)
(posture-task-scale 0.001)
(torque-regular-scale 1.000000e-04)
(stationery-start-finish-regular-scale 1.000000e-04)
(first-diff-square-integration-regular-scale 1.000000e-07)
(second-diff-square-integration-regular-scale 1.000000e-07)
(third-diff-square-integration-regular-scale 1.000000e-07)
(norm-regular-scale-max 1.000000e-05)
(norm-regular-scale-offset 1.000000e-07)

Initialize instance

:robot-env [method]

return robot-environment instance

:variant-joint-list [method]

return $\mathcal{J}_{var}$

:invariant-joint-list [method]

return $\mathcal{J}_{invar}$

:drive-joint-list [method]

return $\mathcal{J}_{drive}$

:num-kin [method]

return $N_{kin} := |\mathcal{T}^{kin\text{-}trg}| = |\mathcal{T}^{kin\text{-}att}|$

:num-contact [method]

return $N_{cnt} := |\mathcal{T}^{cnt\text{-}trg}| = |\mathcal{T}^{cnt\text{-}att}|$

:num-variant-joint [method]

return $N_{var\text{-}joint} := |\mathcal{J}_{var}|$

:num-invariant-joint [method]

return $N_{invar\text{-}joint} := |\mathcal{J}_{invar}|$

:num-drive-joint [method]

return $N_{drive\text{-}joint} := |\mathcal{J}_{drive}|$

:num-posture-joint [method]

return $N_{target\text{-}joint} := |\mathcal{J}_{target}|$

:num-collision [method]

return N_{col} := number of collision check pairs

:dim-config [method]

return $dim(\boldsymbol{q})$

:dim-task [method]

return $dim(\boldsymbol{e})$

:theta-control-vector [method]

return $\boldsymbol{p}_{\theta}$

:cog-control-vector [method]

return $\boldsymbol{p}_{c}$

:ang-moment-control-vector [method]

return $\boldsymbol{p}_{L}$

:wrench-control-vector [method]

return $\boldsymbol{p}_{\hat{w}}$

:torque-control-vector [method]

return $\boldsymbol{p}_{\tau}$

:phi [method]

return $\boldsymbol{\phi}$

:config-vector [method]

return $\boldsymbol{q}$

:set-theta-control-vector *control-vector-new &key (relative? nil)* [method]

Set $\boldsymbol{p}_{\theta}$.

:set-cog-control-vector *control-vector-new &key (relative? nil)* [method]

Set $\boldsymbol{p}_{c}$.

:set-ang-moment-control-vector *control-vector-new &key (relative? nil)* [method]

Set $\boldsymbol{p}_{L}$.

:set-wrench-control-vector *control-vector-new &key (relative? nil)* [method]

Set $\boldsymbol{p}_{\hat{w}}$.

:set-torque-control-vector *control-vector-new &key (relative? nil)* [method]

Set $\boldsymbol{p}_{\tau}$.

:set-phi *phi-new &key (relative? nil)* [method]

Set $\boldsymbol{\phi}$.

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

Set $\boldsymbol{q}$.

:theta *tm &key (diff-order 0)* [method]

return $\boldsymbol{\theta}(t)$ [rad] [m]

:cog *tm &key (diff-order 0)* [method]

return $\boldsymbol{c}(t)$ [m]

:ang-moment *tm &key (diff-order 0)* [method]

return $\boldsymbol{L}(t)$ [kgm^2/s]

:wrench *tm &key (diff-order 0)* [method]

return $\hat{\boldsymbol{w}}(t)$ [N] [Nm]

:torque *tm &key (diff-order 0)* [method]

return $\boldsymbol{\tau}(t)$ [Nm] [N]

:apply-config-to-robot *tm* [method]

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

:kin-target-coords-list *tm* [method]

$$T_m^{kin\text{-}trg} = \{\boldsymbol{p}_m^{kin\text{-}trg}, \boldsymbol{R}_m^{kin\text{-}trg}\} \quad (m = 1, 2, \cdots, N_{kin}) \tag{4.305}$$

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

:kin-attention-coords-list *tm* [method]

$$T_m^{kin\text{-}att} = \{\boldsymbol{p}_m^{kin\text{-}att}, \boldsymbol{R}_m^{kin\text{-}att}\} \quad (m = 1, 2, \cdots, N_{kin}) \tag{4.306}$$

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

:kin-scale-mat-list *tm* [method]

return list of K_{kin}

:contact-target-coords-list *tm* [method]

$$T_m^{cnt\text{-}trg} = \{\boldsymbol{p}_m^{cnt\text{-}trg}, \boldsymbol{R}_m^{cnt\text{-}trg}\} \quad (m = 1, 2, \cdots, N_{cnt}) \tag{4.307}$$

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

:contact-attention-coords-list *tm* [method]

$$T_m^{cnt\text{-}att} = \{\boldsymbol{p}_m^{cnt\text{-}att}, \boldsymbol{R}_m^{cnt\text{-}att}\} \quad (m = 1, 2, \cdots, N_{cnt}) \tag{4.308}$$

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

:contact-constraint-list *tm* [method]

return list of contact-constraint

:wrench-list *tm* [method]

return $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{N_{cnt}}\}$

:force-list *tm* [method]

return $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{N_{cnt}}\}$

:moment-list *tm* [method]

return $\{\boldsymbol{n}_1, \boldsymbol{n}_2, \cdots, \boldsymbol{n}_{N_{cnt}}\}$

:mass [method]

return m [kg]

:mg-vec [method]

return $m\boldsymbol{g}$ [kg m/s^2]

:cog-from-model *&key (update? t)* [method]

return $\boldsymbol{p}_G(\boldsymbol{q})$ [m]

:kinematics-instant-task-value *tm* [method]

$$\bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi}) = \begin{pmatrix} \bar{\boldsymbol{e}}_1^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi}) \\ \bar{\boldsymbol{e}}_2^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi}) \\ \vdots \\ \bar{\boldsymbol{e}}_{\bar{N}_{kin}(t)}^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi}) \end{pmatrix} \in \mathbb{R}^{6\bar{N}_{kin}(t)} \tag{4.309}$$

$$\bar{\boldsymbol{e}}_m^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \begin{pmatrix} \boldsymbol{p}_m^{kin\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}_m^{kin\text{-}att}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \boldsymbol{a}\left(\boldsymbol{R}_m^{kin\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) \boldsymbol{R}_m^{kin\text{-}att}(\boldsymbol{\theta}, \boldsymbol{\phi})^T\right) \end{pmatrix} \in \mathbb{R}^6 \quad (m = 1, 2, \cdots, \bar{N}_{kin}(t)) \tag{4.310}$$

$\boldsymbol{a}(\boldsymbol{R})$ は姿勢行列 $\boldsymbol{R}$ の等価角軸ベクトルを表す.

return $\bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi}) \in \mathbb{R}^{6\bar{N}_{kin}(t)}$

:kinematics-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{kin}(\boldsymbol{p}_\theta, \boldsymbol{\phi}) = \begin{pmatrix} \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t_1), \boldsymbol{\phi}) \\ \vdots \\ \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t_{N_{tm\text{-}kin}}), \boldsymbol{\phi}) \end{pmatrix} \in \mathbb{R}^{6N_{kin}} \quad \left(N_{kin} := \sum_{t=1}^{t_{N_{tm\text{-}kin}}} \bar{N}_{kin}(t)\right) \tag{4.311}$$

return $\boldsymbol{e}^{kin}(\boldsymbol{p}_\theta, \boldsymbol{\phi})$

:eom-trans-instant-task-value *tm* [method]

$$\bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t), \hat{\boldsymbol{w}}(t)) = m\ddot{\boldsymbol{c}} - \left\{ \sum_{\substack{m=1 \\ m \in contact}}^{N_{cnt}} \boldsymbol{f}_m - m\boldsymbol{g} \right\} \tag{4.312}$$

$$= m\ddot{\boldsymbol{c}} - \sum_{\substack{m=1 \\ m \in contact}}^{N_{cnt}} \boldsymbol{f}_m + m\boldsymbol{g} \in \mathbb{R}^3 \tag{4.313}$$

return $\bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t), \hat{\boldsymbol{w}}(t))$

:eom-trans-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{eom\text{-}trans}(\boldsymbol{p}_c, \boldsymbol{p}_{\hat{w}}) \;=\; \begin{pmatrix} \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t_1), \hat{\boldsymbol{w}}(t_1)) \\ \vdots \\ \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}})) \end{pmatrix} \in \mathbb{R}^{3N_{tm\text{-}eom}} \tag{4.314}$$

return $\boldsymbol{e}^{eom\text{-}trans}(\boldsymbol{p}_c, \boldsymbol{p}_{\hat{w}})$

:eom-rot-instant-task-value *tm* [method]

$$\bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi}) \;=\; \dot{\boldsymbol{L}} - \sum_{\substack{m=1 \\ m \in contact}}^{N_{cnt}} \left\{ \left(\boldsymbol{p}_m^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c} \right) \times \boldsymbol{f}_m + \boldsymbol{n}_m \right\} \in \mathbb{R}^3 \tag{4.315}$$

return $\bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})$

:eom-rot-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{eom\text{-}rot}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{p}_L, \boldsymbol{p}_{\hat{w}}, \boldsymbol{\phi}) \;=\; \begin{pmatrix} \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{L}(t_1), \hat{\boldsymbol{w}}(t_1), \boldsymbol{\phi}) \\ \vdots \\ \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{L}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi}) \end{pmatrix} \in \mathbb{R}^{3N_{tm\text{-}eom}} \tag{4.316}$$

return $\boldsymbol{e}^{eom\text{-}rot}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{p}_L, \boldsymbol{p}_{\hat{w}}, \boldsymbol{\phi})$

:cog-instant-task-value *tm* [method]

$$\bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi}) \;=\; \boldsymbol{p}_G(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c} \in \mathbb{R}^3 \tag{4.317}$$

return $\bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})$

:cog-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{cog}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{\phi}) \;=\; \begin{pmatrix} \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{\phi}) \\ \vdots \\ \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi}) \end{pmatrix} \in \mathbb{R}^{3N_{tm\text{-}eom}} \tag{4.318}$$

return $\boldsymbol{e}^{cog}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{\phi})$

:ang-moment-instant-task-value *tm* [method]

$$\bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{\theta}(t), \boldsymbol{L}(t), \boldsymbol{\phi}) \;=\; \boldsymbol{L}(t) - \left\{ \boldsymbol{A}_\theta(\boldsymbol{\theta}(t), \boldsymbol{\phi}(t))\dot{\boldsymbol{\theta}}(t) + \boldsymbol{A}_\phi(\boldsymbol{\theta}(t), \boldsymbol{\phi}(t))\dot{\boldsymbol{\phi}}(t) \right\} \in \mathbb{R}^3 \tag{4.319}$$

本実装では，$\boldsymbol{A}_\theta = \boldsymbol{A}_\phi = \boldsymbol{O}$ という仮定を置く．このとき，タスク関数は次式となる．

$$\bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t)) \;=\; \boldsymbol{L}(t) \in \mathbb{R}^3 \tag{4.320}$$

return $\bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t))$

:ang-moment-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{ang\text{-}moment}(\boldsymbol{p}_L) = \begin{pmatrix} \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t_1)) \\ \vdots \\ \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t_{N_{tm\text{-}eom}})) \end{pmatrix} \in \mathbb{R}^{3N_{tm\text{-}eom}} \tag{4.321}$$

return $\boldsymbol{e}^{ang\text{-}moment}(\boldsymbol{p}_L)$

:posture-instant-task-value *tm* [method]

$$\bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t)) = k_{posture}\left(\boldsymbol{\theta}^{trg}_{posture} - \boldsymbol{\theta}_{posture}\right) \in \mathbb{R}^{N_{posture\text{-}joint}} \tag{4.322}$$

$\boldsymbol{\theta}^{trg}_{posture}, \boldsymbol{\theta}_{posture}$ 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 $\bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t))$

:posture-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{posture}(\boldsymbol{p}_\theta) = \begin{pmatrix} \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t_1)) \\ \vdots \\ \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t_{N_{tm\text{-}kin}})) \end{pmatrix} \in \mathbb{R}^{N_{posture\text{-}joint}N_{tm\text{-}kin}} \tag{4.323}$$

return $\boldsymbol{e}^{posture}(\boldsymbol{p}_\theta)$

:task-value *&key (update? t)* [method]

$$\boldsymbol{e}(\boldsymbol{q}) = \begin{pmatrix} \boldsymbol{e}^{kin}(\boldsymbol{p}_\theta, \boldsymbol{\phi}) \\ \boldsymbol{e}^{eom\text{-}trans}(\boldsymbol{p}_c, \boldsymbol{p}_{\hat{w}}) \\ \boldsymbol{e}^{eom\text{-}rot}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{p}_L, \boldsymbol{p}_{\hat{w}}, \boldsymbol{\phi}) \\ \boldsymbol{e}^{cog}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{\phi}) \\ \boldsymbol{e}^{ang\text{-}moment}(\boldsymbol{p}_\theta, \boldsymbol{p}_L, \boldsymbol{\phi}) \\ \boldsymbol{e}^{trq}(\boldsymbol{p}_\theta, \boldsymbol{p}_{\hat{w}}, \boldsymbol{p}_\tau, \boldsymbol{\phi}) \\ \boldsymbol{e}^{posture}(\boldsymbol{p}_\theta) \end{pmatrix} \tag{4.324}$$

return $\boldsymbol{e}(\boldsymbol{q}) \in \mathbb{R}^{dim(\boldsymbol{e})}$

:kinematics-instant-task-jacobian-with-theta-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_\theta} = \frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_\theta} \tag{4.325}$$

$$\frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{kin}_1(\boldsymbol{\theta}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{kin}_{\bar{N}_{kin}(t)}(\boldsymbol{\theta}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{J}_{1,\theta}(t) \\ \vdots \\ \boldsymbol{J}_{\bar{N}_{kin}(t),\theta}(t) \end{pmatrix} \tag{4.326}$$

$$\frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_\theta} = \frac{\partial}{\partial \boldsymbol{p}_\theta} \boldsymbol{B}_{\theta,n}(t)\boldsymbol{p}_\theta = \boldsymbol{B}_{\theta,n}(t) \tag{4.327}$$

return $\dfrac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{p}_\theta} \in \mathbb{R}^{6N_{kin} \times dim(\boldsymbol{p}_\theta)}$

:kinematics-task-jacobian-with-theta-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{p}_\theta} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t_1),\boldsymbol{\phi}(t_1))}{\partial \boldsymbol{p}_\theta} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t_{N_{tm\text{-}kin}}),\boldsymbol{\phi}(t_{N_{tm\text{-}kin}}))}{\partial \boldsymbol{p}_\theta} \end{pmatrix} \tag{4.328}$$

return $\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{p}_\theta} \in \mathbb{R}^{6N_{kin} \times dim(\boldsymbol{p}_\theta)}$

:kinematics-instant-task-jacobian-with-phi *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t),\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t),\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \tag{4.329}$$

$$\frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t),\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}_1^{kin}(\boldsymbol{\theta}(t),\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}_{\bar{N}_{kin}(t)}^{kin}(\boldsymbol{\theta}(t),\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{J}_{1,\phi}(t) \\ \vdots \\ \boldsymbol{J}_{\bar{N}_{kin}(t),\phi}(t) \end{pmatrix} \tag{4.330}$$

return $\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} \in \mathbb{R}^{6N_{kin} \times dim(\boldsymbol{\phi})}$

:kinematics-task-jacobian-with-phi [method]

$$\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t_1),\boldsymbol{\phi}(t_1))}{\partial \boldsymbol{\phi}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{kin}(\boldsymbol{\theta}(t_{N_{tm\text{-}kin}}),\boldsymbol{\phi}(t_{N_{tm\text{-}kin}}))}{\partial \boldsymbol{\phi}} \end{pmatrix} \tag{4.331}$$

return $\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} \in \mathbb{R}^{6N_{kin} \times dim(\boldsymbol{\phi})}$

:eom-trans-instant-task-jacobian-with-cog-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t),\hat{\boldsymbol{w}}(t))}{\partial \boldsymbol{p}_c} = m\frac{\partial \ddot{\boldsymbol{c}}(t)}{\partial \boldsymbol{p}_c} \tag{4.332}$$

$$= m\frac{\partial}{\partial \boldsymbol{p}_c}\boldsymbol{B}_{c,n-2}(t)\hat{\boldsymbol{D}}_2\boldsymbol{p}_c \tag{4.333}$$

$$= m\boldsymbol{B}_{c,n-2}(t)\hat{\boldsymbol{D}}_2 \tag{4.334}$$

return $\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t),\hat{\boldsymbol{w}}(t))}{\partial \boldsymbol{p}_c} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_c)}$

:eom-trans-task-jacobian-with-cog-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{p}_c} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t_1),\hat{\boldsymbol{w}}(t_1))}{\partial \boldsymbol{p}_c} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t_{N_{tm\text{-}eom}}),\hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}))}{\partial \boldsymbol{p}_c} \end{pmatrix} \tag{4.335}$$

return $\frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{p}_c} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_c)}$

:eom-trans-instant-task-jacobian-with-wrench-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t), \hat{\boldsymbol{w}}(t))}{\partial \boldsymbol{p}_{\hat{w}}} = \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t), \hat{\boldsymbol{w}}(t))}{\partial \hat{\boldsymbol{w}}} \frac{\partial \hat{\boldsymbol{w}}(t)}{\partial \boldsymbol{p}_{\hat{w}}} \quad (4.336)$$

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t), \hat{\boldsymbol{w}}(t))}{\partial \hat{\boldsymbol{w}}} = \begin{pmatrix} -\boldsymbol{I}_3 & \boldsymbol{O}_3 & \cdots & -\boldsymbol{I}_3 & \boldsymbol{O}_3 \end{pmatrix} \quad (4.337)$$

(ただし, $\boldsymbol{p}_m^{cnt\text{-}trg}$が nilの接触については, $\boldsymbol{O}_3$とする)

$$\frac{\partial \hat{\boldsymbol{w}}(t)}{\partial \boldsymbol{p}_{\hat{w}}} = \frac{\partial}{\partial \boldsymbol{p}_{\hat{w}}} \boldsymbol{B}_{\hat{w},n}(t) \boldsymbol{p}_{\hat{w}} = \boldsymbol{B}_{\hat{w},n}(t) \quad (4.338)$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t), \hat{\boldsymbol{w}}(t))}{\partial \boldsymbol{p}_{\hat{w}}} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_{\hat{w}})}$

:eom-trans-task-jacobian-with-wrench-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{p}_{\hat{w}}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t_1), \hat{\boldsymbol{w}}(t_1))}{\partial \boldsymbol{p}_{\hat{w}}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}trans}(\boldsymbol{c}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}))}{\partial \boldsymbol{p}_{\hat{w}}} \end{pmatrix} \quad (4.339)$$

return $\dfrac{\partial \boldsymbol{e}^{eom\text{-}trans}}{\partial \boldsymbol{p}_{\hat{w}}} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_{\hat{w}})}$

:eom-rot-instant-task-jacobian-with-theta-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} = \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_{\theta}} \quad (4.340)$$

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} = \sum_{\substack{m=1 \\ m \in contact}}^{N_{cnt}} \left\{ [\boldsymbol{f}_m(t) \times] \boldsymbol{J}_{m,\theta}^{cnt\text{-}trg}(t) \right\} \quad (4.341)$$

$$\frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_{\theta}} = \frac{\partial}{\partial \boldsymbol{p}_{\theta}} \boldsymbol{B}_{\theta,n}(t) \boldsymbol{p}_{\theta} = \boldsymbol{B}_{\theta,n}(t) \quad (4.342)$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_{\theta})}$

:eom-rot-task-jacobian-with-theta-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_{\theta}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{L}(t_1), \hat{\boldsymbol{w}}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{L}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} \end{pmatrix} \quad (4.343)$$

return $\dfrac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_{\theta}} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_{\theta})}$

:eom-rot-instant-task-jacobian-with-cog-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} = \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{c}} \frac{\partial \boldsymbol{c}(t)}{\partial \boldsymbol{p}_c} \tag{4.344}$$

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{c}} = -\sum_{\substack{m=1 \\ m \in contact}}^{N_{cnt}} [\boldsymbol{f}_m \times] = \left[\left(-\sum_{\substack{m=1 \\ m \in contact}}^{N_{cnt}} \boldsymbol{f}_m \right) \times \right] \tag{4.345}$$

$$\frac{\partial \boldsymbol{c}(t)}{\partial \boldsymbol{p}_c} = \frac{\partial}{\partial \boldsymbol{p}_c} \boldsymbol{B}_{c,n}(t) \boldsymbol{p}_c = \boldsymbol{B}_{c,n}(t) \tag{4.346}$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_c)}$

:eom-rot-task-jacobian-with-cog-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_c} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{L}(t_1), \hat{\boldsymbol{w}}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{L}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} \end{pmatrix} \tag{4.347}$$

return $\dfrac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_c} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_c)}$

:eom-rot-instant-task-jacobian-with-ang-moment-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_L} = \frac{\partial \dot{\boldsymbol{L}}(t)}{\partial \boldsymbol{p}_L} \tag{4.348}$$

$$= \frac{\partial}{\partial \boldsymbol{p}_L} \boldsymbol{B}_{L,n-1}(t) \hat{\boldsymbol{D}}_1 \boldsymbol{p}_L \tag{4.349}$$

$$= \boldsymbol{B}_{L,n-1}(t) \hat{\boldsymbol{D}}_1 \tag{4.350}$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_L} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_L)}$

:eom-rot-task-jacobian-with-ang-moment-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_L} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{L}(t_1), \hat{\boldsymbol{w}}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{p}_L} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{L}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{p}_L} \end{pmatrix} \tag{4.351}$$

return $\dfrac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_L} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_L)}$

:eom-rot-instant-task-jacobian-with-wrench-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\hat{w}}} = \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \hat{\boldsymbol{w}}} \frac{\partial \hat{\boldsymbol{w}}(t)}{\partial \boldsymbol{p}_{\hat{w}}} \qquad (4.3$$

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \hat{\boldsymbol{w}}} = \Big(\big[-\big(\boldsymbol{p}_1^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c}\big) \times \big] \quad -\boldsymbol{I}_3 \quad \cdots \quad \big[-\big(\boldsymbol{p}_{N_{cnt}}^{cnt\text{-}trg}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{c}\big) \times \big] \qquad (4.3$$

(ただし, $\boldsymbol{p}_m^{cnt\text{-}trg}$が nilの接触については, $\boldsymbol{O}_3$とする)

$$\frac{\partial \hat{\boldsymbol{w}}(t)}{\partial \boldsymbol{p}_{\hat{w}}} = \frac{\partial}{\partial \boldsymbol{p}_{\hat{w}}} \boldsymbol{B}_{\hat{w},n}(t) \boldsymbol{p}_{\hat{w}} = \boldsymbol{B}_{\hat{w},n}(t) \qquad (4.3$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\hat{w}}} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_{\hat{w}})}$

:eom-rot-task-jacobian-with-wrench-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_{\hat{w}}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{L}(t_1), \hat{\boldsymbol{w}}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\hat{w}}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{L}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\hat{w}}} \end{pmatrix} \tag{4.355}$$

return $\dfrac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{p}_{\hat{w}}} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_{\hat{w}})}$

:eom-rot-instant-task-jacobian-with-phi *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \tag{4.356}$$

$$= \sum_{\substack{m=1 \\ m \in contact}}^{N_{cnt}} \left\{ [\boldsymbol{f}_m(t) \times] \boldsymbol{J}^{cnt\text{-}trg}_{m,\phi}(t) \right\} \tag{4.357}$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{L}(t), \hat{\boldsymbol{w}}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \in \mathbb{R}^{3 \times dim(\boldsymbol{\phi})}$

:eom-rot-task-jacobian-with-phi [method]

$$\frac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{L}(t_1), \hat{\boldsymbol{w}}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{eom\text{-}rot}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{L}(t_{N_{tm\text{-}eom}}), \hat{\boldsymbol{w}}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \end{pmatrix} \tag{4.358}$$

return $\dfrac{\partial \boldsymbol{e}^{eom\text{-}rot}}{\partial \boldsymbol{\phi}} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{\phi})}$

:cog-instant-task-jacobian-with-theta-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} = \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_{\theta}} \tag{4.359}$$

$$\frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} = \boldsymbol{J}_{G,\theta}(t) \tag{4.360}$$

$$\frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_{\theta}} = \frac{\partial}{\partial \boldsymbol{p}_{\theta}} \boldsymbol{B}_{\theta,n}(t) \boldsymbol{p}_{\theta} = \boldsymbol{B}_{\theta,n}(t) \tag{4.361}$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_{\theta})}$

:cog-task-jacobian-with-theta-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{p}_{\theta}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{p}_{\theta}} \end{pmatrix} \tag{4.362}$$

return $\dfrac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{p}_{\theta}} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_{\theta})}$

:cog-instant-task-jacobian-with-cog-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} = \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{c}} \frac{\partial \boldsymbol{c}(t)}{\partial \boldsymbol{p}_c} \tag{4.363}$$

$$\frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{c}} = -\boldsymbol{I}_3 \tag{4.364}$$

$$\frac{\partial \boldsymbol{c}(t)}{\partial \boldsymbol{p}_c} = \frac{\partial}{\partial \boldsymbol{p}_c} \boldsymbol{B}_{c,n}(t) \boldsymbol{p}_c = \boldsymbol{B}_{c,n}(t) \tag{4.365}$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_c)}$

:cog-task-jacobian-with-cog-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{p}_c} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{p}_c} \end{pmatrix} \tag{4.366}$$

return $\dfrac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{p}_c} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{p}_c)}$

:cog-instant-task-jacobian-with-phi *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \boldsymbol{J}_{G,\phi}(t) \tag{4.367}$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t), \boldsymbol{c}(t), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \in \mathbb{R}^{3 \times dim(\boldsymbol{\phi})}$

:cog-task-jacobian-with-phi [method]

$$\frac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{\phi}} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_1), \boldsymbol{c}(t_1), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{cog}(\boldsymbol{\theta}(t_{N_{tm\text{-}eom}}), \boldsymbol{c}(t_{N_{tm\text{-}eom}}), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \end{pmatrix} \tag{4.368}$$

return $\dfrac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{\phi}} \in \mathbb{R}^{3N_{tm\text{-}eom} \times dim(\boldsymbol{\phi})}$

:ang-moment-instant-task-jacobian-with-ang-moment-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t))}{\partial \boldsymbol{p}_L} = \frac{\partial \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t))}{\partial \boldsymbol{L}} \frac{\partial \boldsymbol{L}(t)}{\partial \boldsymbol{p}_L} \tag{4.369}$$

$$\frac{\partial \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t))}{\partial \boldsymbol{L}} = \boldsymbol{I}_3 \tag{4.370}$$

$$\frac{\partial \boldsymbol{L}(t)}{\partial \boldsymbol{p}_L} = \frac{\partial}{\partial \boldsymbol{p}_L} \boldsymbol{B}_{L,n}(t) \boldsymbol{p}_L = \boldsymbol{B}_{L,n}(t) \tag{4.371}$$

return $\dfrac{\partial \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t))}{\partial \boldsymbol{p}_L} \in \mathbb{R}^{3 \times dim(\boldsymbol{p}_L)}$

:ang-moment-task-jacobian-with-ang-moment-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{ang\text{-}moment}}{\partial \boldsymbol{p}_L} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t_1))}{\partial \boldsymbol{p}_L} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{ang\text{-}moment}(\boldsymbol{L}(t_{N_{tm\text{-}com}}))}{\partial \boldsymbol{p}_L} \end{pmatrix} \tag{4.372}$$

return $\frac{\partial \boldsymbol{e}^{ang\text{-}moment}}{\partial \boldsymbol{p}_L} \in \mathbb{R}^{3N_{tm\text{-}com} \times dim(\boldsymbol{p}_L)}$

:posture-instant-task-jacobian-with-theta-control-vector *tm* [method]

$$\frac{\partial \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t))}{\partial \boldsymbol{p}_\theta} = \frac{\partial \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_\theta} \tag{4.373}$$

$$\left(\frac{\partial \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}}\right)_{i,j} = \begin{cases} -k_{posture} & (\mathcal{J}_{posture,i} = \mathcal{J}_{var,j}) \\ 0 & \text{otherwise} \end{cases} \tag{4.374}$$

$$\frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{p}_\theta} = \frac{\partial}{\partial \boldsymbol{p}_\theta} \boldsymbol{B}_{\theta,n}(t) \boldsymbol{p}_\theta = \boldsymbol{B}_{\theta,n}(t) \tag{4.375}$$

return $\frac{\partial \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t))}{\partial \boldsymbol{p}_\theta} \in \mathbb{R}^{N_{posture\text{-}joint} \times dim(\boldsymbol{p}_\theta)}$

:posture-task-jacobian-with-theta-control-vector [method]

$$\frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{p}_\theta} = \begin{pmatrix} \frac{\partial \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t_1))}{\partial \boldsymbol{p}_\theta} \\ \vdots \\ \frac{\partial \bar{\boldsymbol{e}}^{posture}(\boldsymbol{\theta}(t_{N_{tm\text{-}kin}}))}{\partial \boldsymbol{p}_\theta} \end{pmatrix} \tag{4.376}$$

return $\frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{p}_\theta} \in \mathbb{R}^{N_{posture\text{-}joint} N_{tm\text{-}kin} \times dim(\boldsymbol{p}_\theta)}$

:task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} = \begin{array}{l c} & \begin{array}{cccccc} dim(\boldsymbol{p}_\theta) & dim(\boldsymbol{p}_c) & dim(\boldsymbol{p}_L) & dim(\boldsymbol{p}_{\hat{w}}) & dim(\boldsymbol{p}_\tau) & dim(\boldsymbol{\phi}) \end{array} \\ \begin{array}{l} dim(\boldsymbol{e}^{kin}(\boldsymbol{p}_\theta, \boldsymbol{\phi})) \\ dim(\boldsymbol{e}^{com\text{-}trans}(\boldsymbol{p}_c, \boldsymbol{p}_{\hat{w}})) \\ dim(\boldsymbol{e}^{com\text{-}rot}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{p}_L, \boldsymbol{p}_{\hat{w}}, \boldsymbol{\phi})) \\ dim(\boldsymbol{e}^{cog}(\boldsymbol{p}_\theta, \boldsymbol{p}_c, \boldsymbol{\phi})) \\ dim(\boldsymbol{e}^{ang\text{-}moment}(\boldsymbol{p}_\theta, \boldsymbol{p}_L, \boldsymbol{\phi})) \\ dim(\boldsymbol{e}^{trq}(\boldsymbol{p}_\theta, \boldsymbol{p}_{\hat{w}}, \boldsymbol{p}_\tau, \boldsymbol{\phi})) \\ dim(\boldsymbol{e}^{posture}(\boldsymbol{p}_\theta)) \end{array} & \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{p}_\theta} & & & & & \frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\phi}} \\ & \frac{\partial \boldsymbol{e}^{com\text{-}trans}}{\partial \boldsymbol{p}_c} & & \frac{\partial \boldsymbol{e}^{com\text{-}trans}}{\partial \boldsymbol{p}_{\hat{w}}} & & \\ \frac{\partial \boldsymbol{e}^{com\text{-}rot}}{\partial \boldsymbol{p}_\theta} & \frac{\partial \boldsymbol{e}^{com\text{-}rot}}{\partial \boldsymbol{p}_c} & \frac{\partial \boldsymbol{e}^{com\text{-}rot}}{\partial \boldsymbol{p}_L} & \frac{\partial \boldsymbol{e}^{com\text{-}rot}}{\partial \boldsymbol{p}_{\hat{w}}} & & \frac{\partial \boldsymbol{e}^{com\text{-}rot}}{\partial \boldsymbol{\phi}} \\ \frac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{p}_\theta} & \frac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{p}_c} & & & & \frac{\partial \boldsymbol{e}^{cog}}{\partial \boldsymbol{\phi}} \\ & & \frac{\partial \boldsymbol{e}^{ang\text{-}moment}}{\partial \boldsymbol{p}_L} & & & \\ & & & & & \\ \frac{\partial \boldsymbol{e}^{posture}}{\partial \boldsymbol{p}_\theta} & & & & & \end{pmatrix} \end{array} \tag{4.377}$$

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

:theta-max-vector *&key (update? nil)* [method]

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

:theta-min-vector *&key (update? nil)* [method]

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

:theta-instant-inequality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{\theta}_{min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_{max} \tag{4.378}$$
$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \boldsymbol{\theta} \geq \begin{pmatrix} \boldsymbol{\theta}_{min} \\ -\boldsymbol{\theta}_{max} \end{pmatrix} \tag{4.379}$$
$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{\theta}} \boldsymbol{\theta} \geq \bar{\boldsymbol{d}}_{\boldsymbol{\theta}} \tag{4.380}$$

return $\boldsymbol{C}_{\boldsymbol{\theta}} := \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{2N_{var\text{-}joint} \times N_{var\text{-}joint}}$

:theta-instant-inequality-constraint-vector *&key (update? nil)* [method]

return $\bar{\boldsymbol{d}}_{\boldsymbol{\theta}} := \begin{pmatrix} \boldsymbol{\theta}_{min} \\ -\boldsymbol{\theta}_{max} \end{pmatrix} \in \mathbb{R}^{2N_{var\text{-}joint}}$

:theta-control-vector-inequality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{C}_{\boldsymbol{\theta}} \boldsymbol{\theta} \geq \bar{\boldsymbol{d}}_{\boldsymbol{\theta}} \tag{4.381}$$
$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{p_\theta}} \boldsymbol{p}_\theta \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_\theta}} \tag{4.382}$$

差分形式で表すと次式となる．

$$\boldsymbol{C}_{\boldsymbol{p_\theta}} (\boldsymbol{p}_\theta + \Delta \boldsymbol{p}_\theta) \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_\theta}} \tag{4.383}$$
$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{p_\theta}} \Delta \boldsymbol{p}_\theta \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_\theta}} - \boldsymbol{C}_{\boldsymbol{p_\theta}} \boldsymbol{p}_\theta \tag{4.384}$$
$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{p_\theta}} \Delta \boldsymbol{p}_\theta \geq \boldsymbol{d}_{\boldsymbol{p_\theta}} \tag{4.385}$$

return $\boldsymbol{C}_{p_\theta}$

:theta-control-vector-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{p_\theta} := \bar{\boldsymbol{d}}_{\boldsymbol{p_\theta}} - \boldsymbol{C}_{\boldsymbol{p_\theta}} \boldsymbol{p}_\theta$

:cog-max-vector *&key (update? nil)* [method]

return $\boldsymbol{c}_{max} \in \mathbb{R}^3$ [m]

:cog-instant-inequality-constraint-matrix *&key (update? nil)* [method]

$$-\boldsymbol{c}_{max} \leq \boldsymbol{c} \leq \boldsymbol{c}_{max} \tag{4.386}$$
$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \boldsymbol{c} \geq \begin{pmatrix} -\boldsymbol{c}_{max} \\ -\boldsymbol{c}_{max} \end{pmatrix} \tag{4.387}$$
$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{c}} \boldsymbol{c} \geq \bar{\boldsymbol{d}}_{\boldsymbol{c}} \tag{4.388}$$

return $\boldsymbol{C}_{\boldsymbol{c}} := \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{6 \times 3}$

:cog-instant-inequality-constraint-vector *&key (update? nil)* [method]

return $\bar{\boldsymbol{d}}_{\boldsymbol{c}} := \begin{pmatrix} -\boldsymbol{c}_{max} \\ -\boldsymbol{c}_{max} \end{pmatrix} \in \mathbb{R}^6$

:cog-control-vector-inequality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{C_c c} \geq \bar{\boldsymbol{d}}_{\boldsymbol{c}} \tag{4.389}$$

$$\Leftrightarrow \quad \boldsymbol{C_{p_c} p}_c \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_c}} \tag{4.390}$$

差分形式で表すと次式となる.

$$\boldsymbol{C_{p_c}}(\boldsymbol{p}_c + \Delta \boldsymbol{p}_c) \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_c}} \tag{4.391}$$

$$\Leftrightarrow \quad \boldsymbol{C_{p_c}} \Delta \boldsymbol{p}_c \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_c}} - \boldsymbol{C_{p_c} p}_c \tag{4.392}$$

$$\Leftrightarrow \quad \boldsymbol{C_{p_c}} \Delta \boldsymbol{p}_c \geq \boldsymbol{d_{p_c}} \tag{4.393}$$

return $\boldsymbol{C}_{p_c}$

:cog-control-vector-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{p_c} := \bar{\boldsymbol{d}}_{\boldsymbol{p_c}} - \boldsymbol{C_{p_c} p}_c$

:ang-moment-max-vector *&key (update? nil)* [method]

return $\boldsymbol{L}_{max} \in \mathbb{R}^3$ [kgm^2/s]

:ang-moment-instant-inequality-constraint-matrix *&key (update? nil)* [method]

$$-\boldsymbol{L}_{max} \leq \boldsymbol{L} \leq \boldsymbol{L}_{max} \tag{4.394}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \boldsymbol{L} \geq \begin{pmatrix} -\boldsymbol{L}_{max} \\ -\boldsymbol{L}_{max} \end{pmatrix} \tag{4.395}$$

$$\Leftrightarrow \quad \boldsymbol{C_L L} \geq \bar{\boldsymbol{d}}_{\boldsymbol{L}} \tag{4.396}$$

return $\boldsymbol{C_L} := \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{6 \times 3}$

:ang-moment-instant-inequality-constraint-vector *&key (update? nil)* [method]

return $\bar{\boldsymbol{d}}_{\boldsymbol{L}} := \begin{pmatrix} -\boldsymbol{L}_{max} \\ -\boldsymbol{L}_{max} \end{pmatrix} \in \mathbb{R}^6$

:ang-moment-control-vector-inequality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{C_L L} \geq \bar{\boldsymbol{d}}_{\boldsymbol{L}} \tag{4.397}$$

$$\Leftrightarrow \quad \boldsymbol{C_{p_L} p}_L \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_L}} \tag{4.398}$$

差分形式で表すと次式となる.

$$\boldsymbol{C_{p_L}}(\boldsymbol{p}_L + \Delta \boldsymbol{p}_L) \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_L}} \tag{4.399}$$

$$\Leftrightarrow \quad \boldsymbol{C_{p_L}} \Delta \boldsymbol{p}_L \geq \bar{\boldsymbol{d}}_{\boldsymbol{p_L}} - \boldsymbol{C_{p_L} p}_L \tag{4.400}$$

$$\Leftrightarrow \quad \boldsymbol{C_{p_L}} \Delta \boldsymbol{p}_L \geq \boldsymbol{d_{p_L}} \tag{4.401}$$

return $\boldsymbol{C}_{p_L}$

:ang-moment-control-vector-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{p_L} := \bar{\boldsymbol{d}}_{\boldsymbol{p_L}} - \boldsymbol{C_{p_L} p}_L$

:wrench-instant-inequality-constraint-matrix *&key (update? t)* [method]

接触レンチ $\boldsymbol{w} \in \mathbb{R}^6$ 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．

$$\boldsymbol{C}_w \boldsymbol{w} \geq \boldsymbol{d}_w \tag{4.402}$$

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

$$\boldsymbol{C}_{w,m} \boldsymbol{w}_m \geq \boldsymbol{d}_{w,m} \quad (m = 1, 2, \cdots, N_{cnt}) \tag{4.403}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C}_{w,1} & & & \\ & \boldsymbol{C}_{w,2} & & \\ & & \ddots & \\ & & & \boldsymbol{C}_{w,N_{cnt}} \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_{N_{cnt}} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_{w,1} \\ \boldsymbol{d}_{w,2} \\ \vdots \\ \boldsymbol{d}_{w,N_{cnt}} \end{pmatrix} \tag{4.404}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\hat{w}} \hat{\boldsymbol{w}} \geq \boldsymbol{d}_{\hat{w}} \tag{4.405}$$

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

:wrench-instant-inequality-constraint-vector *&key (update? nil)* [method]

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

:wrench-control-vector-inequality-constraint-matrix *&key (update? nil)* [method]

$$\boldsymbol{C}_{\hat{\boldsymbol{w}}} \hat{\boldsymbol{w}} \geq \bar{\boldsymbol{d}}_{\hat{\boldsymbol{w}}} \tag{4.406}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \boldsymbol{p}_{\hat{w}} \geq \bar{\boldsymbol{d}}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \tag{4.407}$$

差分形式で表すと次式となる．

$$\boldsymbol{C}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} (\boldsymbol{p}_{\hat{w}} + \Delta \boldsymbol{p}_{\hat{w}}) \geq \bar{\boldsymbol{d}}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \tag{4.408}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \Delta \boldsymbol{p}_{\hat{w}} \geq \bar{\boldsymbol{d}}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} - \boldsymbol{C}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \boldsymbol{p}_{\hat{w}} \tag{4.409}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \Delta \boldsymbol{p}_{\hat{w}} \geq \boldsymbol{d}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \tag{4.410}$$

return $\boldsymbol{C}_{p_{\hat{w}}}$

:wrench-control-vector-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d}_{p_{\hat{w}}} := \bar{\boldsymbol{d}}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} - \boldsymbol{C}_{\boldsymbol{p}_{\hat{\boldsymbol{w}}}} \boldsymbol{p}_{\hat{w}}$

:torque-control-vector-inequality-constraint-matrix [method]

todo

:torque-control-vector-inequality-constraint-vector [method]

todo

:phi-max-vector *&key (update? nil)* [method]

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

:phi-min-vector *&key (update? nil)* [method]

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

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

$$\boldsymbol{\phi}_{min} \leq \boldsymbol{\phi} + \Delta\boldsymbol{\phi} \leq \boldsymbol{\phi}_{max} \tag{4.411}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta\boldsymbol{\phi} \geq \begin{pmatrix} \boldsymbol{\phi}_{min} - \boldsymbol{\phi} \\ -(\boldsymbol{\phi}_{max} - \boldsymbol{\phi}) \end{pmatrix} \tag{4.412}$$

$$\Leftrightarrow \quad \boldsymbol{C}_{\boldsymbol{\phi}} \Delta\boldsymbol{\phi} \geq \boldsymbol{d}_{\boldsymbol{\phi}} \tag{4.413}$$

return $\boldsymbol{C}_{\boldsymbol{\phi}} := \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{2N_{invar\text{-}joint} \times N_{invar\text{-}joint}}$

:phi-inequality-constraint-vector *&key (update? t)* [method]

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

:config-inequality-constraint-matrix [method]

$$\begin{cases} \boldsymbol{C}_{p_\theta} \Delta\boldsymbol{p}_\theta \geq \boldsymbol{d}_{p_\theta} \\ \boldsymbol{C}_{p_c} \Delta\boldsymbol{p}_c \geq \boldsymbol{d}_{p_c} \\ \boldsymbol{C}_{p_L} \Delta\boldsymbol{p}_L \geq \boldsymbol{d}_{p_L} \\ \boldsymbol{C}_{p_{\hat{w}}} \Delta\boldsymbol{p}_{\hat{w}} \geq \boldsymbol{d}_{p_{\hat{w}}} \\ \boldsymbol{C}_{p_\tau} \Delta\boldsymbol{p}_\tau \geq \boldsymbol{d}_{p_\tau} \\ \boldsymbol{C}_{\phi} \Delta\boldsymbol{\phi} \geq \boldsymbol{d}_{\phi} \end{cases} \tag{4.414}$$

$$\Leftrightarrow \quad \begin{pmatrix} \boldsymbol{C}_{p_\theta} & & & & & \\ & \boldsymbol{C}_{p_c} & & & & \\ & & \boldsymbol{C}_{p_L} & & & \\ & & & \boldsymbol{C}_{p_{\hat{w}}} & & \\ & & & & \boldsymbol{C}_{p_\tau} & \\ & & & & & \boldsymbol{C}_{\phi} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{p}_\theta \\ \Delta\boldsymbol{p}_c \\ \Delta\boldsymbol{p}_L \\ \Delta\boldsymbol{p}_{\hat{w}} \\ \Delta\boldsymbol{p}_\tau \\ \Delta\boldsymbol{\phi} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{d}_{p_\theta} \\ \boldsymbol{d}_{p_c} \\ \boldsymbol{d}_{p_L} \\ \boldsymbol{d}_{p_{\hat{w}}} \\ \boldsymbol{d}_{p_\tau} \\ \boldsymbol{d}_{\phi} \end{pmatrix} \tag{4.415}$$

$$\Leftrightarrow \quad \boldsymbol{C} \Delta\boldsymbol{q} \geq \boldsymbol{d} \tag{4.416}$$

return $\boldsymbol{C}$

:config-inequality-constraint-vector [method]

return $\boldsymbol{d}$

:config-equality-constraint-matrix *&key (update? nil)* [method]

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

:config-equality-constraint-vector *&key (update? t)* [method]

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

:stationery-start-finish-regular-matrix *&key (update? nil)* [method]

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

:differential-square-integration-regular-matrix *&key (diff-order 1)* [method]

return $\boldsymbol{W}_{sqr,d} \in \mathbb{R}^{dim(\boldsymbol{q}) \times dim(\boldsymbol{q})}$

:first-differential-square-integration-regular-matrix *&key (update? nil)* [method]

return $\boldsymbol{W}_{sqr,1} \in \mathbb{R}^{dim(\boldsymbol{q}) \times dim(\boldsymbol{q})}$

:second-differential-square-integration-regular-matrix *&key (update? nil)* [method]

return $\boldsymbol{W}_{sqr,2} \in \mathbb{R}^{dim(\boldsymbol{q}) \times dim(\boldsymbol{q})}$

:third-differential-square-integration-regular-matrix *&key (update? nil)* [method]

return $\boldsymbol{W}_{sqr,3} \in \mathbb{R}^{dim(\boldsymbol{q}) \times dim(\boldsymbol{q})}$

:regular-matrix [method]

$$\boldsymbol{W}_{reg} := \min(k_{max}, \|\boldsymbol{e}\|^2 + k_{off})\boldsymbol{I} + k_{stat}\boldsymbol{W}_{stat} + \sum_{d=1}^{3} k_{sqr,d}\boldsymbol{W}_{sqr,d} \tag{4.417}$$

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

:regular-vector [method]

$$\boldsymbol{v}_{reg} := k_{stat}\boldsymbol{W}_{stat}\boldsymbol{q} + \sum_{d=1}^{3} k_{sqr,d}\boldsymbol{W}_{sqr,d}\boldsymbol{q} \tag{4.418}$$

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

:update-collision-inequality-constraint [method]

Not implemented yet.

:update-viewer *&key (start-time (send _theta-bst :start-time))* [method]
(finish-time (send _theta-bst :finish-time))
(delta-time (/ (- finish-time start-time) 100.0))

Update viewer.

:print-setting-information [method]

Print setting information.

:print-status [method]

Print status.

:play-animation *&key (robot-env)* [method]
(start-time (send _theta-bst :start-time))
(finish-time (send _theta-bst :finish-time))
(delta-time (/ (- finish-time start-time) 100.0))
(loop? t)
(visualize-callback-func)

Play motion animation.

:generate-graph *&key (start-time (send _theta-bst :start-time))* [method]
(finish-time (send _theta-bst :finish-time))
(delta-time (/ (- finish-time start-time) 100.0))
(data-dirname /tmp/bspline-dynamic-config-task)
(graph-filename /tmp/bspline-dynamic-config-task/graph.pdf)

Generate graph from configuration and task trajectory.

:generate-robot-state-list *&key (robot-env _robot-env)* [method]
(start-time (send _theta-bst :start-time))
(finish-time (send _theta-bst :finish-time))
(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.

4.4 離散的な幾何目標に対する逆運動学計算

4.4.1 離散的な幾何目標に対する逆運動学計算の理論

min/max 関数の微分可能関数近似

minimum/maximum 関数

$$F_{min}(\boldsymbol{x}; f_1, \cdots, f_K) \overset{\text{def}}{=} \min(f_1(\boldsymbol{x}), \cdots, f_K(\boldsymbol{x})) \tag{4.419}$$
$$F_{max}(\boldsymbol{x}; f_1, \cdots, f_K) \overset{\text{def}}{=} \max(f_1(\boldsymbol{x}), \cdots, f_K(\boldsymbol{x})) \tag{4.420}$$

を連続かつ微分可能な関数で近似した smooth minimum/maximum 関数として，次式を用いることができる [12].

$$\mathcal{S}_\alpha(\boldsymbol{x}; f_1, \cdots, f_K) \overset{\text{def}}{=} \frac{\sum_{k=1}^{K} f_k(\boldsymbol{x}) e^{\alpha f_k(\boldsymbol{x})}}{\sum_{k=1}^{K} e^{\alpha f_k(\boldsymbol{x})}} \tag{4.421}$$

この関数は以下の性質をもつ．

$$\alpha \to -\inf \quad \text{のとき} \quad \mathcal{S}_\alpha \to F_{min} \tag{4.422}$$
$$\alpha \to \inf \quad \text{のとき} \quad \mathcal{S}_\alpha \to F_{max} \tag{4.423}$$

離散的な目標に対するタスク関数の微分可能関数近似

タスク関数として $\boldsymbol{e}_1(\boldsymbol{q}), \cdots, \boldsymbol{e}_K(\boldsymbol{q}) \in \mathbb{R}^{N_e}$ が与えられているときに，これらのタスク関数のいずれかをゼロにするコンフィギュレーション $\boldsymbol{q} \in \mathbb{R}^{N_q}$ を求める問題を考える．複数個の目標位置のいずれかにリーチングする逆運動学問題などがこの問題に含まれる．

この問題は次式で表される．

$$\boldsymbol{e}_k(\boldsymbol{q}) = \boldsymbol{0} \quad (k \text{ は } 1, \cdots, K \text{ のいずれか}) \tag{4.424}$$

これは次式と同値である．

$$\boldsymbol{e}_{min}(\boldsymbol{q}) = \boldsymbol{0} \tag{4.425}$$
$$\text{where} \quad \boldsymbol{e}_{min}(\boldsymbol{q}) \overset{\text{def}}{=} \underset{\boldsymbol{e}_k \in \mathcal{E}}{\arg\min} \|\boldsymbol{e}_k(\boldsymbol{q})\|^2 \in \mathbb{R}^{N_e} \tag{4.426}$$
$$\mathcal{E} \overset{\text{def}}{=} \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_K\} \tag{4.427}$$

[12] https://en.wikipedia.org/wiki/Smooth_maximum

タスク関数 $\boldsymbol{e}_{min}(\boldsymbol{q})$ のヤコビ行列 $\frac{\partial \boldsymbol{e}_{min}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ が導出できれば，第 1 章の定式化により最適化計算を行うことでコンフィギュレーション $\boldsymbol{q}$ を求めることができる．しかし，$\boldsymbol{e}_{min}(\boldsymbol{q})$ は一般に，最小の $\boldsymbol{e}_k$ が切り替わる点において微分不可能であり，ヤコビ行列を求めることができない．

式 (4.421) では，$f_k(\boldsymbol{x}) \in \mathbb{R}\ (k=1,\cdots,K)$ の $\frac{e^{\alpha f_k(\boldsymbol{x})}}{\sum_{k=1}^{K} e^{\alpha f_k(\boldsymbol{x})}}$ による重み付けした和をとることで，min/max の微分可能関数近似を得ている．この近似をスカラ値関数からベクトル値関数へと拡張して，$\boldsymbol{e}_{min}(\boldsymbol{q})$ を次式の微分可能関数で近似する．

$$\hat{\boldsymbol{e}}_{min}(\boldsymbol{q}) \stackrel{\text{def}}{=} \frac{1}{\sum_{\boldsymbol{e}_k \in \mathcal{E}} \exp(-\alpha \|\boldsymbol{e}_k(\boldsymbol{q})\|^2)} \sum_{\boldsymbol{e}_k \in \mathcal{E}} \exp(-\alpha \|\boldsymbol{e}_k(\boldsymbol{q})\|^2) \boldsymbol{e}_k(\boldsymbol{q}) \in \mathbb{R}^{N_e} \tag{4.428}$$

α は正の定数で大きいほど近似精度が増す．タスク関数 $\hat{\boldsymbol{e}}_{min}(\boldsymbol{q})$ のヤコビ行列 $\frac{\partial \hat{\boldsymbol{e}}_{min}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ は，解析的に導出可能である．

contact-invariant-optimization における微分可能関数近似 (参考)

contact-invariant-optimization の論文 [13] の 4.1 節では，minimum 関数を含むタスク関数が以下のように近似されている．

$$\hat{\boldsymbol{e}}_{min}(\boldsymbol{q}) \stackrel{\text{def}}{=} \frac{1}{\sum_{\boldsymbol{e}_k \in \mathcal{E}} \eta(\boldsymbol{e}_k(\boldsymbol{q}))} \sum_{\boldsymbol{e}_k \in \mathcal{E}} \eta(\boldsymbol{e}_k(\boldsymbol{q})) \boldsymbol{e}_k(\boldsymbol{q}) \in \mathbb{R}^{N_e} \tag{4.429}$$

$$\text{where} \quad \eta(\boldsymbol{e}_k(\boldsymbol{q})) = \frac{1}{1+\beta \|\boldsymbol{e}_k(\boldsymbol{q})\|^2} \in \mathbb{R} \tag{4.430}$$

β は正の定数で，論文では 10^4 としている．これは，式 (4.428) における $\exp(-\alpha\|\boldsymbol{e}_k(\boldsymbol{q})\|^2)$ を $\eta(\boldsymbol{e}_k(\boldsymbol{q}))$ で置き換えたものである．

LogSumExp による微分可能関数近似 (参考)

式 (4.419) , 式 (4.420) の minimum/maximum 関数を連続かつ微分可能な関数で近似した smooth minimum/maximum 関数として，LogSumExp 関数を用いることができる [14]．

$$LSE_{\varepsilon}(\boldsymbol{x}; f_1, \cdots, f_K) \stackrel{\text{def}}{=} \frac{\log\left(\sum_{k=1}^{K} \exp(\varepsilon f_k(\boldsymbol{x}))\right)}{\varepsilon} \tag{4.431}$$

ε が負のとき minimum 関数，正のとき maximum 関数の近似となり，絶対値が大きいほど近似精度が増す．

この関数は，重み付け和の形式ではないため，式 (4.428) のようにスカラ値関数からベクトル値関数へ拡張することができない．

タスク関数のノルム二乗として表される最適化の目的関数

$$F(\boldsymbol{q}) \stackrel{\text{def}}{=} \min_{\boldsymbol{e}_k \in \mathcal{E}} \|\boldsymbol{e}_k(\boldsymbol{q})\|^2 \in \mathbb{R} \tag{4.432}$$

は，次の $\hat{F}(\boldsymbol{q})$ として近似できる．

$$\hat{F}(\boldsymbol{q}) \approx \frac{\log\left(\sum_{\boldsymbol{e}_k \in \mathcal{E}} \exp(-\varepsilon \|\boldsymbol{e}_k(\boldsymbol{q})\|^2)\right)}{-\varepsilon} \tag{4.433}$$

式 (4.431) の ε を改めて $-\varepsilon$ と置き直した．ε が大きいほど近似精度が増す．

[13] Discovery of complex behaviors through contact-invariant optimization, I. Mordatch, et. al., ACM Transactions on Graphics 31.4, 43, 2012.

[14] `https://en.wikipedia.org/wiki/Smooth_maximum`

近似目的関数 $\hat{F}(\boldsymbol{q})$ の勾配は次式で表される．

$$\frac{\partial \hat{F}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \frac{\sum_{\boldsymbol{e}_k \in \mathcal{E}} 2\varepsilon \exp(-\varepsilon \|\boldsymbol{e}_k(\boldsymbol{q})\|^2) \left(\frac{\partial \boldsymbol{e}_k(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T \boldsymbol{e}_k(\boldsymbol{q})}{\varepsilon \sum_{\boldsymbol{e}_k \in \mathcal{E}} \exp(-\varepsilon \|\boldsymbol{e}_k(\boldsymbol{q})\|^2)} \tag{4.434}$$

近似目的関数 $\hat{F}(\boldsymbol{q})$ のヘッセ行列も解析的に導出可能である．(タスク関数を考える場合，そのヤコビ行列が求まれば，第 1 章のように目的関数のヘッセ行列は導出可能である．しかし，今回のように目的関数を直接扱う場合は，そのヘッセ行列を陽に導出する必要がある．)

4.4.2 離散的な幾何目標に対する逆運動学計算の実装

discrete-kinematics-configuration-task [class]

:super **instant-configuration-task**
:slots (_smooth-alpha α)

離散的な幾何目標を扱えるように拡張された瞬時コンフィギュレーション $\boldsymbol{q}^{(l)}$ と瞬時タスク関数 $\boldsymbol{e}^{(l)}(\boldsymbol{q}^{(l)})$ のクラス．

離散的な幾何目標とは，kin-target-coords-list や kin-attention-coords-list として目標位置姿勢や着目位置姿勢が複数ペア与えられ，それらのいずれかが成り立てば良いという制約のことを指す．離散的な幾何目標のタスク関数に含まれる min 関数を微分可能関数で近似することで，タスク関数のヤコビ行列を求める．

:init *&rest args &key (smooth-alpha 20.0) &allow-other-keys* [method]

Initialize instance

:kinematics-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{kin}(\boldsymbol{q}) = \boldsymbol{e}^{kin}(\boldsymbol{\theta}, \boldsymbol{\phi}) \tag{4.435}$$

$$= \begin{pmatrix} \boldsymbol{e}^{kin}_1(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \boldsymbol{e}^{kin}_2(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \vdots \\ \boldsymbol{e}^{kin}_{N_{kin}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \end{pmatrix} \tag{4.436}$$

$$\boldsymbol{e}^{kin}_m(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathop{\arg\min}_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2 \in \mathbb{R}^6 \quad (m = 1, 2, \cdots, N_{kin}) \tag{4.437}$$

$$\text{where} \quad \mathcal{E}^{kin}_m = \{\boldsymbol{e}^{kin}_{m,i} \mid i = 1, 2, \cdots, N_{kin\text{-}dis,m}\} \tag{4.438}$$

$$\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) = K_{kin} \begin{pmatrix} \boldsymbol{p}^{kin\text{-}trg}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{p}^{kin\text{-}att}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ \boldsymbol{a}\left(\boldsymbol{R}^{kin\text{-}trg}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) \boldsymbol{R}^{kin\text{-}att}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})^T\right) \end{pmatrix} \in \mathbb{R}^6 \quad (i = 1, 2, \cdots, N_{kin\text{-}dis,m}) \tag{4.439}$$

$\boldsymbol{a}(\boldsymbol{R})$ は姿勢行列 $\boldsymbol{R}$ の等価角軸ベクトルを表す．$\boldsymbol{e}^{kin}_m(\boldsymbol{\theta}, \boldsymbol{\phi})$ を次式で近似する．

$$\boldsymbol{e}^{kin}_m(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{\sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} \exp(-\alpha \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2)} \sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} \exp(-\alpha \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2) \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})$$

$$\in \mathbb{R}^6 \quad (m = 1, 2, \cdots, N_{kin}) \tag{4.440}$$

α は正の定数で大きいほど近似精度が増す.

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

:kinematics-task-jacobian-with-theta [method]

$$\frac{\partial \boldsymbol{e}^{kin}}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{kin}_1}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{e}^{kin}_2}{\partial \boldsymbol{\theta}} \\ \vdots \\ \frac{\partial \boldsymbol{e}^{kin}_{N_{kin}}}{\partial \boldsymbol{\theta}} \end{pmatrix} \tag{4.441}$$

ここで,

$$\begin{aligned} \boldsymbol{e}^{kin}_m(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \frac{1}{\sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} \exp(-\alpha \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2)} \sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} \exp(-\alpha \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2) \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ &= u(\boldsymbol{\theta}, \boldsymbol{\phi}) \boldsymbol{v}(\boldsymbol{\theta}, \boldsymbol{\phi}) \end{aligned} \tag{4.442}$$

$$\text{where} \quad u(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{\sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} \exp(-\alpha \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2)} \in \mathbb{R} \tag{4.443}$$

$$\boldsymbol{v}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} \exp(-\alpha \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2) \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) \in \mathbb{R}^6 \tag{4.444}$$

であるから,

$$\frac{\partial \boldsymbol{e}^{kin}_m(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} = \boldsymbol{v}(\boldsymbol{\theta}, \boldsymbol{\phi}) \left(\frac{\partial u(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \right)^T + u(\boldsymbol{\theta}, \boldsymbol{\phi}) \left(\frac{\partial \boldsymbol{v}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \right) \tag{4.445}$$

$$\frac{\partial u(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} = \frac{\sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} 2\alpha A \left(\frac{\partial \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\left\{ \sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} A \right\}^2} \in \mathbb{R}^{N_{var\text{-}joint}} \tag{4.446}$$

$$\frac{\partial \boldsymbol{v}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} = \sum_{\boldsymbol{e}^{kin}_{m,i} \in \mathcal{E}^{kin}_m} A \left\{ -2\alpha \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})^T \left(\frac{\partial \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \right) + \left(\frac{\partial \boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} \right) \right\} \in \mathbb{R}^{6 \times N_{var\text{-}joint}} \tag{4.447}$$

$$\frac{\partial \boldsymbol{e}^{kin}_{m,i}}{\partial \boldsymbol{\theta}} = K_{kin} \left\{ \boldsymbol{J}^{kin\text{-}trg}_{\theta,m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \boldsymbol{J}^{kin\text{-}att}_{\theta,m,i}(\boldsymbol{\theta}, \boldsymbol{\phi}) \right\} \tag{4.448}$$

$$(m = 1, 2, \cdots, N_{kin}, \quad i = 1, 2, \cdots, N_{kin\text{-}dis,m})$$

ただし,

$$A = \exp(-\alpha \|\boldsymbol{e}^{kin}_{m,i}(\boldsymbol{\theta}, \boldsymbol{\phi})\|^2) \tag{4.449}$$

とした.

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

4.5 ボディ表面のコンフィギュレーションとタスク関数

4.5.1 ボディ表面の連続関数近似

line [class]

:super **propertied-object**
:slots (pvert)
(nvert)

:nearest-point-distance *p* [method]

get the nearest point and distance to the nearest point

face [class]

:super **polygon**
:slots (normal)
(distance)
(convexp)
(edges)
(vertices)
(model-normal)
(model-distance)
(holes)
(mbody)
(primitive-face)
(id)

:nearest-point-distance *point* [method]

get the nearest point and distance to the nearest point

:neighbor-face-point-distance *pos &key (exclude-faces)* [method]

get the nearest point and distance for all neighbor faces

:serach-neighbor-face-within-distance *pos &key (dist 0.0)* [method]
(dist-max 500.0)
(depth 1)
(depth-max 3)
(exclude-faces)
&aux
(neighbor-face-point-distance (send self :neighbor-face-point-dis

search faces within distance

:neighbor-face-distance *pos &key (dist-max 500.0)* [method]
(depth-max 3)
&aux
(face-dist-list (send self :serach-neighbor-face-within-distance pos :dist-max dist-
(faces (remove self (remove-duplicates (mapcar #'(lambda (x) (elt x 0)) face-di

get face and minimum distance within distance

faceset [class]

:super **cascaded-coords**
:slots (rot)
(pos)
(parent)
(descendants)
(worldcoords)
(manager)
(changed)
(box)
(faces)
(edges)
(vertices)
(model-vertices)

:set-projection-mode *mode* [method]

set projection mode

:project *pos* [method]

project point to body surface. projected point is the nearest point on the body.

:project-with-bt-collision *pos &aux (sphere (send self :get :sphere-for-projection))* [method]

project point to body surface by using bullet collision function.

:project-with-distance *pos* [method]

project point to body surface by computing the distance.

:calc-normal-tangent *pos &key (flip-normal? nil)* [method]
&aux
(smooth-dist 50.0)
(normal (scale (if flip-normal? -1 1) (calc-smooth-normal self pos :smooth-dist sm
*(tangent1 (cond ((eps= (norm (v*normal #f(1.0 0.0 0.0))) 0.0) (normalize-vector*
*(tangent2 (v*normal tangent1))*

calculate one smooth normal vector and two tanget vectors on the surface point.

calc-smooth-normal *body0 pos &key (smooth-dist 20.0)* [function]

Calculate smooth normal by weighted average of neighbor normals. Average is calculaged by using slearp.

slearp *ratio v0 v1 &key (dot-thre 0.9995)* [function]
&aux
(dot (v. v0 v1))

Spherical linear interpolation. Implemented based on Python sample in `https://en.wikipedia.org/wiki/Slerp`

slearp2 *ratio v0 v1 &aux (n (v*v0 v1)) (theta0 (acos (v. v0 v1))) (theta (*ratio theta0))* [function]

Spherical linear interpolation. Implemented based on 3D rotation matrix.

interpolate-one-to-zero *x &optional (a 1.0) &aux (x/a (/ x a)) (x/a-1 (- x/a 1.0))* [function]

Interpolate from 1 to 0 with piecewise quadratic function.

$$f(x) = \begin{cases} -2\left(\frac{x}{a}\right)^2 + 1 & (x \le 0.5) \\ 2\left(\frac{x}{a} - 1\right)^2 & (x \ge 0.5) \end{cases} \tag{4.450}$$

calc-weighted-average-with-interpolation *value-list weight-list &key (interpolate-func #'midpoint) &aux (ave-value (car value-list)) (accumulated-weight (float (car weight-list) (weight) (value)* [function]

Calculate weighted average from interpolation function.

make-ellipsoid *&optional (a 100.0) (b 100.0) (c 200.0) &key (theta-divide-num 20) (phi-divide-num (*2 theta-divide-num)) (half? nil)* [function]

Make ellipsoid body.

generate-surface-configuration-task *&key ((:link link0)) ((:body body0) (convex-hull-3d (remove-duplicates (flatten (send-all (s (kin-target-coords) (initial-surface-pos (cond ((functionp kin-target-coords) (send (funcal (delta-u 0.05) (flip-normal? nil) (normal-task-scale 2.0) (norm-regular-scale-max 1.000000e+10) (norm-regular-scale-offset 1.0) (config-task)* [function]

generate surface-configuration-task instance from link.

calc-weighted-average *value-list weight-list* [function]

4.5.2 ボディ表面

surface-configuration-task [class]

:super **propertied-object**

:slots (_body body)

(_surface-coords cascaded-coords of surface point $\boldsymbol{p} \in \mathbb{R}^3$ [mm])

(_pos-normal-tangent $\boldsymbol{p}, \boldsymbol{\xi}, (\boldsymbol{\zeta}, \boldsymbol{\eta}) \in \mathbb{R}^3$ surface position, normal, and tangents)

(_dim-u $dim(\boldsymbol{u}) := 2$)

(_dim-config $dim(\boldsymbol{q})$)

(_dim-task $dim(\boldsymbol{e})$)

(_normal-task-scale k_{nrm})

(_norm-regular-scale-max k_{max})

(_norm-regular-scale-offset k_{off})

(_kin-target-coords $\mathcal{T}^{kin\text{-}trg}$)
(_delta-u $\Delta \boldsymbol{u}_{limit} \in \mathbb{R}^2$)
(_flip-normal? whether to flip the normal or not)

ボディ表面上の着目点を扱うためのコンフィギュレーションとタスク関数のクラス．

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

ボディ表面の着目点の位置 $\boldsymbol{p} \in \mathbb{R}^3$ は，二次元コンフィギュレーション $\boldsymbol{q} \in \mathbb{R}^2$ で表現できる．しかし，単一の二次元パラメータをボディ表面上の点に網羅的に対応させる写像を得ることは困難である．したがって，本クラスではコンフィギュレーション $\boldsymbol{q}$ を明示的に保持することはしない．代わりに，保持している物体表面上の点 $\boldsymbol{p} \in \mathbb{R}^3$ の周りに，局所的な座標系 $(\boldsymbol{\zeta}, \boldsymbol{\eta}, \boldsymbol{\xi})$ を構築し，この座標系で定義された局所二次元コンフィギュレーション $\boldsymbol{u}$ を構築する．$\boldsymbol{\zeta}, \boldsymbol{\eta}$ は点 $\boldsymbol{p}$ におけるボディの接ベクトルを表し，$\boldsymbol{\xi}$ は法線ベクトルを表す．このコンフィギュレーションは，$\boldsymbol{u} = \boldsymbol{0}$ で $\boldsymbol{p}$ に対応し，$\boldsymbol{u}$ の第 1,2 要素がそれぞれ，接ベクトル $\boldsymbol{\zeta}, \boldsymbol{\eta} \in \mathbb{R}^3$ 方向に対応するものとする．

:init *&key (name)* [method]
((:body tmp-body))
(initial-surface-pos)
(kin-target-coords)
(delta-u 0.02)
(flip-normal? nil)
(normal-task-scale 1)
(norm-regular-scale-max 1.000000e+10)
(norm-regular-scale-offset 1.0)
&allow-other-keys

Initialize instance

:body [method]

return body instance

:dim-config [method]

return $dim(\boldsymbol{q}) := 2$

:dim-task [method]

return $dim(\boldsymbol{e}) := 3$

:config-vector [method]

Not supported because this class does not have configuration $\boldsymbol{q}$ explicitly.

return $\boldsymbol{q}$

:set-config *delta-config &key (relative? t)* [method]

Set $\boldsymbol{q}$.

The surface point $\boldsymbol{p}$ is updated from $\boldsymbol{q}$.

:surface-pos [method]

get surface point

return $\boldsymbol{p} \in \mathbb{R}^3$

:surface-coords *&key (surface-pos (elt _pos-normal-tangent 0))*
(surface-normal (elt _pos-normal-tangent 1)) [method]

get surface coords. z-axis is parallel with surface normal.

:kin-target-coords [method]

get kinematics target coords.

:position-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{pos}(\boldsymbol{q}) = \boldsymbol{p}^{kin\text{-}trg} - \boldsymbol{p}(\boldsymbol{u}) \tag{4.451}$$

return $\boldsymbol{e}^{pos}(\boldsymbol{q}) \in \mathbb{R}^3$

:normal-task-value *&key (update? t)* [method]

$$\boldsymbol{e}^{nrm}(\boldsymbol{q}) = 1 - \boldsymbol{\xi}^{kin\text{-}trg,T}\boldsymbol{\xi}(\boldsymbol{u}) \tag{4.452}$$

$\boldsymbol{\xi}^{kin\text{-}trg}, \boldsymbol{\xi}(\boldsymbol{u})$ はそれぞれ，目標点，着目点の法線ベクトルを表す.

return $\boldsymbol{e}^{nrm}(\boldsymbol{q}) \in \mathbb{R}$

:task-value *&key (update? t)* [method]

return $\boldsymbol{e}(\boldsymbol{q}) := \begin{pmatrix} \boldsymbol{e}^{pos}(\boldsymbol{q}) \\ k_{nrm}\boldsymbol{e}^{nrm}(\boldsymbol{q}) \end{pmatrix} \in \mathbb{R}^4$

:position-task-jacobian-with-u [method]

$$\frac{\partial \boldsymbol{e}^{pos}}{\partial \boldsymbol{u}} = -\begin{pmatrix} \boldsymbol{\zeta} & \boldsymbol{\eta} \end{pmatrix} \tag{4.453}$$

return $\frac{\partial \boldsymbol{e}^{pos}}{\partial \boldsymbol{u}} \in \mathbb{R}^{3\times 2}$

:normal-task-jacobian-with-u [method]

$$\frac{\partial \boldsymbol{e}^{nrm}}{\partial \boldsymbol{u}} = -\boldsymbol{\xi}^{kin\text{-}trg,T}\frac{\partial \boldsymbol{\xi}(\boldsymbol{u})}{\partial \boldsymbol{u}} \tag{4.454}$$

return $\frac{\partial \boldsymbol{e}^{nrm}}{\partial \boldsymbol{u}} \in \mathbb{R}^{1\times 2}$

:task-jacobian [method]

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} = \begin{pmatrix} \frac{\partial \boldsymbol{e}^{pos}}{\partial \boldsymbol{u}} \\ k_{nrm}\frac{\partial \boldsymbol{e}^{nrm}}{\partial \boldsymbol{u}} \end{pmatrix} \tag{4.455}$$

return $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} \in \mathbb{R}^{4\times 2}$

:delta-u-limit-vector *&key (update? nil)*
(delta-u1 _delta-u)
(delta-u2 _delta-u) [method]

get trust region of $\boldsymbol{u}$

return $\Delta\boldsymbol{u}_{limit} \in \mathbb{R}^2$

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

$$
\begin{aligned}
& -\Delta \boldsymbol{u}_{limit} \leq \Delta \boldsymbol{u} \leq \Delta \boldsymbol{u}_{limit} && (4.456)\\
\Leftrightarrow \quad & \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta \boldsymbol{u} \geq \begin{pmatrix} -\Delta \boldsymbol{u}_{limit} \\ -\Delta \boldsymbol{u}_{limit} \end{pmatrix} && (4.457)\\
\Leftrightarrow \quad & \boldsymbol{C_u} \Delta \boldsymbol{u} \geq \boldsymbol{d_u} && (4.458)
\end{aligned}
$$

return $\boldsymbol{C_u} := \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{4\times 2}$

:u-inequality-constraint-vector *&key (update? t)* [method]

return $\boldsymbol{d_u} := \begin{pmatrix} -\Delta \boldsymbol{u}_{limit} \\ -\Delta \boldsymbol{u}_{limit} \end{pmatrix} \in \mathbb{R}^4$

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

$$\boldsymbol{C} \Delta \boldsymbol{q} \geq \boldsymbol{d} \tag{4.459}$$

return $\boldsymbol{C} := \boldsymbol{C_u}$

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

return $\boldsymbol{d} := \boldsymbol{d_u}$

:config-equality-constraint-matrix *&key (update? nil)* [method]

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

:config-equality-constraint-vector *&key (update? t)* [method]

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

:regular-matrix [method]

$$\boldsymbol{W}_{reg} := \min(k_{max}, \|\boldsymbol{e}\|^2 + k_{off})\boldsymbol{I} \tag{4.460}$$

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

:regular-vector [method]

$$\boldsymbol{v}_{reg} := \boldsymbol{0} \tag{4.461}$$

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

:distance-vector *config-task* [method]

$$\boldsymbol{d} := \boldsymbol{p}(\boldsymbol{u}) - \boldsymbol{p}_{other}(\boldsymbol{u}_{other}) \tag{4.462}$$

Distance $\boldsymbol{d}$ is used for sqp-msc (sqp with multiple solution candidates).

return $\boldsymbol{d} \in \mathbb{R}^3$

:distance-jacobian [method]

$$\boldsymbol{J}_d := \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{u}} = \begin{pmatrix} \boldsymbol{\zeta} & \boldsymbol{\eta} \end{pmatrix} \tag{4.463}$$

Distance jacobian $\boldsymbol{J}_d$ is used for sqp-msc (sqp with multiple solution candidates).

return $\boldsymbol{J}_d \in \mathbb{R}^{3\times2}$

:update-viewer [method]

Update viewer.

:print-status [method]

Print status.

4.5.3 関節とボディ表面

joint-surface-configuration-task [class]

:super **compound-configuration-task**

:slots (_instant-config-task instant-configuration-task instance)

(_surface-kin-contact-list list of (surface-config-task, kin-attention-name, contact-atten

(_instant-config-distance-scale scale for distance calculation of instant-configuration-ta

多関節機構の先に付いたボディ表面上の着目点を扱うためのコンフィギュレーションとタスク関数のクラス.

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

:init *&key (instant-config-task)* [method]
(surface-kin-contact-list)
(instant-config-distance-scale 1.0)

Initialize instance

:instant-config-task [method]

return instant-configuration-task instance.

:surface-kin-contact-list [method]

return surface-kin-contact-list, which is list of (surface-config-task, kin-attention-name, contact-attention-name).

:set-config *config-new &key (relative? nil)* [method]
&aux
(surface-config-task)
(config-idx)

Set $\boldsymbol{q}$.

:task-value *&key (update? t)* [method]

return $\boldsymbol{e}(\boldsymbol{q})$

:task-jacobian *&aux (instant-jacobi (send _instant-config-task :task-jacobian)) (kin-rotation-type-list (send _instant-config-task :get-val '_kin-rotation-type-list)) (config-idx)* [method]

return $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}}$

:distance-vector *config-task* [method]

return distance $\boldsymbol{d}$, which is used for sqp-msc (sqp with multiple solution candidates).

:distance-jacobian [method]

return distance jacobian $\boldsymbol{J}_d$, which is used for sqp-msc (sqp with multiple solution candidates).

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 *&rest 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 $\boldsymbol{a}_i \in \mathbb{R}^3$

:pos [method]

Return joint position. Represented in world coordinates.

return $\boldsymbol{p}_i \in \mathbb{R}^3$

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)

:centroid-with-fixed-child-links [method]

return $\boldsymbol{p}_{cog,k} \in \mathbb{R}^3$ [mm]

:weight-with-fixed-child-links [method]

return $m \in \mathbb{R}$ [g]

:mg [method]

return $mg = \|m\boldsymbol{g}\| \in \mathbb{R}$ [N]

:mg-vec [method]

return $m\boldsymbol{g} \in \mathbb{R}^3$ [N]

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 *&key* *(union-joint-list)* [method]
(move-target)
(joint-list (mapcar #'(lambda (mt) (send-all (send self :link-list (send mt :pa
(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).

find-fixed-child-links *l &key joint-list* [function]

set-mass-property-with-fixed-child-links *robot* [function]

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 $T_{cnt\text{-}pre}$)

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 *&key (name contact)*
(contact-pre-offset 100) [method]

Initialize instance

:contact-coords *&rest args* [method]

return $T_{cnt} := \{\boldsymbol{p}_{cnt}, \boldsymbol{R}_{cnt}\}$

:contact-pre-coords *&rest args* [method]

return $T_{cnt\text{-}pre} := \{\boldsymbol{p}_{cnt\text{-}pre}, \boldsymbol{R}_{cnt\text{-}pre}\}$

:set-from-face *&key (face)*
(margin 150.0) [method]

set coords and min/max joint angle from face.

look-at-contact [class]

:super **cascaded-link**
:slots (_contact-coords T_{cnt})

ある点を注視するためのカメラ座標を表す仮想の関節・リンク構造.

:init *&key (name look-at)*
(target-pos (float-vector 0 0 0))
(camera-axis :z)
(angle-of-view 30.0) [method]

Initialize instance

:contact-coords *&rest args* [method]

return $T_{cnt} := \{\boldsymbol{p}_{cnt}, \boldsymbol{R}_{cnt}\}$

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 $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{N_C}\}$)
(_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 $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{N_C}\}$ 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 *&key* *(robot)* [method]
(contact-list)
(root-virtual-mode :6dof)
(root-virtual-joint-class-list)
(root-virtual-joint-axis-list)
(root-virtual-joint-min-angle-list)
(root-virtual-joint-max-angle-list)

Initialize instance

:dissoc-root-virtual [method]

dissoc root virtual parent/child structure.

:init-pose [method]

set zero joint angle.

:robot *&rest args* [method]

return $\mathcal{R}$

:robot-with-root-virtual *&rest args* [method]

return $\hat{\mathcal{R}}$

:contact-list *&rest args* [method]

return $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{N_C}\}$

:contact *name &rest args* [method]

return $\mathcal{C}_i$

:variant-joint-list *&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-li
(rotation-axis (cond ((atom move-target) t) (t (make-list
(thre (cond ((atom move-target) 1) (t (make-list (length r
(rthre (cond ((atom move-target) (deg2rad 1)) (t (make-li
(collision-avoidance-link-pair :nil)
(collision-distance-limit 10.0)
(obstacles)
(min-loop)
(root-virtual-mode :fix)
(root-virtual-joint-min-angle-list)
(root-virtual-joint-max-angle-list)
(joint-angle-margin 0.0)
(posture-joint-list)
(posture-joint-angle-list)
(target-posture-scale 0.001)

(norm-regular-scale-max 0.01)
(norm-regular-scale-offset 1.000000e-07)
(pre-process-func)
(post-process-func)
&allow-other-keys

Solve inverse kinematics problem with sqp optimization. ;; target-coords, move-target, rotation-axis, translation-axis ;; -¿ both list and atom OK. target-coords : 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. link-list : List of links to control. When the target-coords is list, this should be a list of lists. move-target : 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 : :x :y :z for constraint along the x, y, z axis. :xy :yz :zx for plane. Default is t. rotation-axis : 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).

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-trajectory-optmotiongen *target-coords-list &key* *(stop 50)* [method]
(move-target-list)
(debug-view)
(revert-if-fail t)
(transform-coords-list :nil)
(translation-axis-list :nil)
(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)
(root-virtual-joint-min-angle-list)
(root-virtual-joint-max-angle-list)
(joint-angle-margin 0.0)
(posture-joint-list (make-list (length targe
(posture-joint-angle-list (make-list (length
(norm-regular-scale-max 0.001)
(norm-regular-scale-offset 1.000000e-07)
(adjacent-regular-scale 0.0)
(pre-process-func)
(post-process-func)
&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 &rest 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* 引数に対応する接触座標を表す仮想の関節・リンク構造.

:init *&key (target-coords)* [method]
(translation-axis)
(rotation-axis)
(transform-coords target-coords)
(name (send target-coords :name))

Initialize instance

:contact-coords *&rest args* [method]

return $T_{cnt} := \{\boldsymbol{p}_{cnt}, \boldsymbol{R}_{cnt}\}$

ik-arg-axis->axis-list *ik-arg-axis* [function]

Convert translation-axis / rotatoin-axis to axis list.

generate-contact-ik-arg-from-rect-face *&key (rect-face)* [function]
(name (send rect-face :name))
(margin (or (send rect-face :get :margin) 0))

Generate contact-ik-arg instance from rectangle face.

generate-contact-ik-arg-from-line-segment *&key (line-seg)* [function]
(name (send line-seg :name))
(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 *&key (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* の位置を $\boldsymbol{p}_m \in \mathbb{R}^3$，*drive-joint-list* の関節角度ベクトルを $\boldsymbol{\psi} \in \mathbb{R}^{(N_{drive\text{-}joint})}$ として，次式を満たすヤコビ行列 $\boldsymbol{J}_m$ を返す．

$$\boldsymbol{J}_m = \begin{pmatrix} \boldsymbol{j}_m^{(1)} & \boldsymbol{j}_m^{(2)} & \cdots & \boldsymbol{j}_m^{(N_{drive\text{-}joint})} \end{pmatrix} \tag{5.1}$$

$$\boldsymbol{j}_m^{(i)} = \begin{cases} \bar{\boldsymbol{j}}_m^{(i)} & \text{接触リンクが } i \text{ 番目の駆動関節変位} \psi_i \text{に依存している場合} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{5.2}$$

$\bar{\boldsymbol{j}}_m^{(i)}$ は基礎ヤコビ行列の列ベクトルで次式で表される．
ψ_i が回転関節の場合

$$\bar{\boldsymbol{j}}_m^{(i)} = \begin{pmatrix} \boldsymbol{a}_{\psi_i} \times (\boldsymbol{p}_m - \boldsymbol{p}_{\psi_i}) \\ \boldsymbol{a}_{\psi_i} \end{pmatrix} \tag{5.3}$$

ψ_i が直動関節の場合

$$\bar{\boldsymbol{j}}_m^{(i)} = \begin{pmatrix} \boldsymbol{a}_{\psi_i} \\ \boldsymbol{0} \end{pmatrix} \tag{5.4}$$

$\boldsymbol{a}_{\psi_i}, \boldsymbol{p}_{\psi_i} \in \mathbb{R}^3$ は i 番目の関節の回転軸ベクトルと位置である．

return $\boldsymbol{J}_m \in \mathbb{R}^{6 \times N_{drive\text{-}joint}}$

get-contact-torque *&key (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．

$$\boldsymbol{\tau}^{cnt} = \sum_{m=1}^{N_{cnt}} \boldsymbol{J}_m^T \boldsymbol{w}_m \tag{5.5}$$

$\boldsymbol{w}_m$ は m 番目の接触部位で受ける接触レンチである．

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

get-contact-torque-jacobian *&key* *(robot)* [function]
(joint-list)
(drive-joint-list)
(wrench-list)
(contact-target-coords-list)
(contact-attention-coords-list)

以下では，$\boldsymbol{p}_B^A$ は A から B へ向かう位置ベクトルをワールド座標系で表記したものとする．A,B は，*drive-joint-list* の関節位置 ψ_i，*joint-list* の関節位置 θ_i，*contact-coords* の位置 m のいずれかを指す．

$$\begin{aligned}
\frac{\partial \boldsymbol{\tau}^{cnt}}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \sum_{m=1}^{N_{cnt}} \boldsymbol{J}_m^T \boldsymbol{w}_m & (5.6)\\
&= \frac{\partial}{\partial \boldsymbol{\theta}} \sum_{m=1}^{N_{cnt}} \begin{pmatrix} \boldsymbol{j}_m^{(1)} & \boldsymbol{j}_m^{(2)} & \cdots & \boldsymbol{j}_m^{(N_{drive\text{-}joint})} \end{pmatrix}^T \boldsymbol{w}_m & (5.7)\\
&= \sum_{m=1}^{N_{cnt}} \frac{\partial}{\partial \boldsymbol{\theta}} \begin{pmatrix} \boldsymbol{w}_m^T \boldsymbol{j}_m^{(1)} \\ \boldsymbol{w}_m^T \boldsymbol{j}_m^{(2)} \\ \vdots \\ \boldsymbol{w}_m^T \boldsymbol{j}_m^{(N_{drive\text{-}joint})} \end{pmatrix} & (5.8)\\
&= \sum_{m=1}^{N_{cnt}} \left[\boldsymbol{w}_m^T \frac{\partial}{\partial \theta_j} \boldsymbol{j}_m^{(i)} \right]_{i,j} \quad (i = 1, 2, \cdots, N_{drive\text{-}joint},\quad j = 1, 2, \cdots, N_{joint}) & (5.9)
\end{aligned}$$

したがって，各接触力によるトルクのヤコビ行列の各要素は次式で得られる．
ψ_i が回転関節の場合

$$\begin{aligned}
\boldsymbol{w}_m^T \frac{\partial}{\partial \theta_j} \boldsymbol{j}_m^{(i)} &= \begin{pmatrix} \boldsymbol{f}_m \\ \boldsymbol{n}_m \end{pmatrix}^T \frac{\partial}{\partial \theta_j} \begin{pmatrix} \boldsymbol{a}_{\psi_i} \times \boldsymbol{p}_m^{\psi_i} \\ \boldsymbol{a}_{\psi_i} \end{pmatrix} & (5.10)\\
&= \boldsymbol{f}_m^T \frac{\partial}{\partial \theta_j} \left(\boldsymbol{a}_{\psi_i} \times \boldsymbol{p}_m^{\psi_i} \right) + \boldsymbol{n}_m^T \frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} & (5.11)\\
&= \boldsymbol{f}_m^T \left\{ \left(\frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} \right) \times \boldsymbol{p}_m^{\psi_i} + \boldsymbol{a}_{\psi_i} \times \left(\frac{\partial}{\partial \theta_j} \boldsymbol{p}_m - \frac{\partial}{\partial \theta_j} \boldsymbol{p}_{\psi_i} \right) \right\} + \boldsymbol{n}_m^T \frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} & (5.12)
\end{aligned}$$

ψ_i が直動関節の場合

$$\begin{aligned}
\boldsymbol{w}_m^T \frac{\partial}{\partial \theta_j} \boldsymbol{j}_m^{(i)} &= \begin{pmatrix} \boldsymbol{f}_m \\ \boldsymbol{n}_m \end{pmatrix}^T \frac{\partial}{\partial \theta_j} \begin{pmatrix} \boldsymbol{a}_{\psi_i} \\ \boldsymbol{0} \end{pmatrix} & (5.13)\\
&= \boldsymbol{f}_m^T \frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} & (5.14)
\end{aligned}$$

$\frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i}$ *(drive-jnt-axis-derivative)*，$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{\psi_i}$ *(drive-jnt-pos-derivative)* は以下のように計算される．

(A) 関節 θ_j が関節 ψ_i よりもルートリンクに近いとき，もしくは関節 θ_j と関節 ψ_i が同一のとき，

(I) 関節 θ_j が回転関節のとき，回転系での基礎方程式から，

$$\frac{\partial}{\partial\theta_j}\boldsymbol{a}_{\psi_i} = \boldsymbol{a}_{\theta_j} \times \boldsymbol{a}_{\psi_i} \tag{5.15}$$

$$\frac{\partial}{\partial\theta_j}\boldsymbol{p}_{\psi_i} = \boldsymbol{a}_{\theta_j} \times \boldsymbol{p}_{\psi_i}^{\theta_j} \tag{5.16}$$

(II) 関節 θ_j が直動関節のとき

$$\frac{\partial}{\partial\theta_j}\boldsymbol{a}_{\psi_i} = \boldsymbol{0} \tag{5.17}$$

$$\frac{\partial}{\partial\theta_j}\boldsymbol{p}_{\psi_i} = \boldsymbol{a}_{\theta_j} \tag{5.18}$$

(B) *(A)* でないとき，つまり
関節 ψ_i が関節 θ_j よりもルートリンクに近いとき，もしくは，ルートリンクから関節 θ_j までの間とルートリンクから関節 ψ_i までの間に共通の関節が存在しないとき，関節 θ_j の変化は関節 ψ_i の位置，回転軸のベクトルに影響を与えないため，

$$\frac{\partial}{\partial\theta_j}\boldsymbol{a}_{\psi_i} = \boldsymbol{0} \tag{5.19}$$

$$\frac{\partial}{\partial\theta_j}\boldsymbol{p}_{\psi_i} = \boldsymbol{0} \tag{5.20}$$

$\frac{\partial}{\partial\theta_j}\boldsymbol{p}_m$ *(contact-pos-derivative)* は以下のように計算される．

(a) 関節 θ_j の変位が $\boldsymbol{p}_m$ に影響を与えるとき *(*このパターンは *contact-target-coords* が仮想関節の先にが設置されている場合などに発生する*)*

(i) 関節 θ_j が回転関節のとき

$$\frac{\partial}{\partial\theta_j}\boldsymbol{p}_m = \boldsymbol{a}_{\theta_j} \times \boldsymbol{p}_m^{\theta_j} \tag{5.21}$$

(ii) 関節 θ_j が直動関節のとき

$$\frac{\partial}{\partial\theta_j}\boldsymbol{p}_m = \boldsymbol{a}_{\theta_j} \tag{5.22}$$

(b) *(a)* でないとき，つまり
関節 θ_j の変位が $\boldsymbol{p}_m$ に影響を与えないとき

$$\frac{\partial}{\partial\theta_j}\boldsymbol{p}_m = \boldsymbol{0} \tag{5.23}$$

return $\frac{\partial\boldsymbol{\tau}^{cnt}}{\partial\boldsymbol{\theta}} \in \mathbb{R}^{N_{drive\text{-}joint} \times N_{joint}}$

get-link-jacobian-for-gravity-torque *&key* *(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 $\boldsymbol{p}_{cog,k} \in \mathbb{R}^3$，*drive-joint-list* の関節角度ベクトルを $\boldsymbol{\psi} \in \mathbb{R}^{N_{drive\text{-}joint}}$ として，次式を満たすヤコビ行列 $\boldsymbol{J}_{cog,k}$ を返す．

$$\dot{\boldsymbol{p}}_{cog,k} = \boldsymbol{J}_{cog,k}\dot{\boldsymbol{\psi}} = \sum_{i=1}^{N_k} \boldsymbol{j}_{cog,k}^{(i)}\dot{\psi}_i \tag{5.24}$$

$$\boldsymbol{J}_{cog,k} = \begin{pmatrix} \boldsymbol{j}_{cog,k}^{(1)} & \boldsymbol{j}_{cog,k}^{(2)} & \cdots & \boldsymbol{j}_{cog,k}^{(N_{drive\text{-}joint})} \end{pmatrix} \tag{5.25}$$

$$\boldsymbol{j}_{cog,k}^{(i)} = \begin{cases} \bar{\boldsymbol{j}}_{cog,k}^{(i)} & \textit{gravity-link}\text{が } i \text{ 番目の駆動関節変位}\psi_i\text{に依存している場合} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{5.26}$$

$\bar{\boldsymbol{j}}_{cog,k}^{(i)}$ は基礎ヤコビ行列の列ベクトルで次式で表される．
ψ_i が回転関節の場合

$$\bar{\boldsymbol{j}}_{cog,k}^{(i)} = \boldsymbol{a}_{\psi_i} \times (\boldsymbol{p}_{cog,k} - \boldsymbol{p}_{\psi_i}) \tag{5.27}$$

ψ_i が直動関節の場合

$$\bar{\boldsymbol{j}}_{cog,k}^{(i)} = \boldsymbol{a}_{\psi_i} \tag{5.28}$$

$\boldsymbol{a}_{\psi_i}, \boldsymbol{p}_{\psi_i} \in \mathbb{R}^3$ は i 番目の関節の回転軸ベクトルと位置である．

return $\boldsymbol{J}_{cog,k} \in \mathbb{R}^{3 \times N_{drive\text{-}joint}}$

get-gravity-torque *&key* *(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．

$$\boldsymbol{\tau}^{grav} = \left(\sum_{k=1}^{N_{gravity\text{-}link}} m_k \boldsymbol{J}_{cog,k}^T \right) \boldsymbol{g} \tag{5.29}$$

m_k は k 番目のリンクの質量である．

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

get-gravity-torque-jacobian *&key* *(robot)* [function]
(joint-list)
(drive-joint-list)
(gravity-link-list)

$$\begin{aligned}
\frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \theta_j} \sum_{k=1}^{N_{gravity\text{-}link}} \boldsymbol{J}_{cog,k}^T m_k \boldsymbol{g} && (5.30) \\
&= \frac{\partial}{\partial \theta_j} \sum_{k=1}^{N_{gravity\text{-}link}} \begin{pmatrix} \boldsymbol{j}_{cog,k}^{(1)} & \boldsymbol{j}_{cog,k}^{(2)} & \cdots & \boldsymbol{j}_{cog,k}^{(N_{drive\text{-}joint})} \end{pmatrix}^T m_k \boldsymbol{g} && (5.31) \\
&= \sum_{k=1}^{N_{gravity\text{-}link}} \frac{\partial}{\partial \theta_j} \begin{pmatrix} m_k \boldsymbol{g}^T \boldsymbol{j}_{cog,k}^{(1)} \\ m_k \boldsymbol{g}^T \boldsymbol{j}_{cog,k}^{(2)} \\ \vdots \\ m_k \boldsymbol{g}^T \boldsymbol{j}_{cog,k}^{(N_{drive\text{-}joint})} \end{pmatrix} && (5.32) \\
&= \sum_{k=1}^{N_{gravity\text{-}link}} \left[m_k \boldsymbol{g}^T \frac{\partial}{\partial \theta_j} \boldsymbol{j}_{cog,k}^{(i)} \right]_{i,j} \quad (i = 1, 2, \cdots, N_{drive\text{-}joint}, \quad j = 1, 2, \cdots, N_{joint}) && (5.33)
\end{aligned}$$

したがって，各リンクの重力によるトルクのヤコビ行列の各要素は次式で得られる．
k 番目の *gravity-link-list* が i 番目の駆動関節変位 ψ_i に依存していない場合

$$m_k \boldsymbol{g}^T \frac{\partial}{\partial \theta_j} \boldsymbol{j}_{cog,k}^{(i)} = 0 \tag{5.34}$$

k 番目の *gravity-link-list* が i 番目の駆動関節変位 ψ_i に依存していて，ψ_i が回転関節の場合

$$m_k \boldsymbol{g}^T \frac{\partial}{\partial \theta_j} \boldsymbol{j}^{(i)}_{cog,k} = m_k \boldsymbol{g}^T \frac{\partial}{\partial \theta_j} \left(\boldsymbol{a}_{\psi_i} \times \boldsymbol{p}^{\psi_i}_{cog,k} \right) \tag{5.35}$$

$$= m_k \boldsymbol{g}^T \left\{ \left(\frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} \right) \times \boldsymbol{p}^{\psi_i}_{cog,k} + \boldsymbol{a}_{\psi_i} \times \left(\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{cog,k} - \frac{\partial}{\partial \theta_j} \boldsymbol{p}_{\psi_i} \right) \right\} \tag{5.36}$$

k 番目の *gravity-link-list* が i 番目の駆動関節変位 ψ_i に依存していて，ψ_i が直動関節の場合

$$m_k \boldsymbol{g}^T \frac{\partial}{\partial \theta_j} \boldsymbol{j}^{(i)}_{cog,k} = m_k \boldsymbol{g}^T \frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} \tag{5.37}$$

$\frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i}$ *(drive-jnt-axis-derivative)*，$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{\psi_i}$ *(drive-jnt-pos-derivative)* は以下のように計算される．

(A) 関節 θ_j が関節 ψ_i よりもルートリンクに近いとき，もしくは関節 θ_j と関節 ψ_i が同一のとき，

(I) 関節 θ_j が回転関節のとき，回転系での基礎方程式から，

$$\frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} = \boldsymbol{a}_{\theta_j} \times \boldsymbol{a}_{\psi_i} \tag{5.38}$$

$$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{\psi_i} = \boldsymbol{a}_{\theta_j} \times \boldsymbol{p}^{\theta_j}_{\psi_i} \tag{5.39}$$

(II) 関節 θ_j が直動関節のとき

$$\frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} = \boldsymbol{0} \tag{5.40}$$

$$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{\psi_i} = \boldsymbol{a}_{\theta_j} \tag{5.41}$$

(B) *(A)* でないとき，つまり

関節 ψ_i が関節 θ_j よりもルートリンクに近いとき，もしくは，ルートリンクから関節 θ_j までの間とルートリンクから関節 ψ_i までの間に共通の関節が存在しないとき，関節 θ_j の変化は関節 ψ_i の位置，回転軸のベクトルに影響を与えないため，

$$\frac{\partial}{\partial \theta_j} \boldsymbol{a}_{\psi_i} = \boldsymbol{0} \tag{5.42}$$

$$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{\psi_i} = \boldsymbol{0} \tag{5.43}$$

$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{cog,k}$ *(centroid-derivative)* は以下のように計算される．

(a) k 番目の *gravity-link-list* が j 番目の関節変位 θ_j に依存しているとき

(i) 関節 θ_j が回転関節のとき

$$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{cog,k} = \boldsymbol{a}_{\theta_j} \times \boldsymbol{p}^{\theta_j}_{cog,k} \tag{5.44}$$

(ii) 関節 θ_j が直動関節のとき

$$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{cog,k} = \boldsymbol{a}_{\theta_j} \tag{5.45}$$

(b) *(a)* でないとき，つまり

k 番目の *gravity-link-list* が j 番目の関節変位 θ_j に依存していないとき

$$\frac{\partial}{\partial \theta_j} \boldsymbol{p}_{cog,k} = \boldsymbol{0} \tag{5.46}$$

return $\frac{\partial \boldsymbol{\tau}^{grav}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{N_{drive\text{-}joint} \times N_{joint}}$