逆動力学解析マニュアル

DhaibaWorks V2

バージョン r21851 対応

2020 年 8 月 24 日作成

著作　産業技術総合研究所　遠藤　維

更新情報

2020/8/24　初版作成

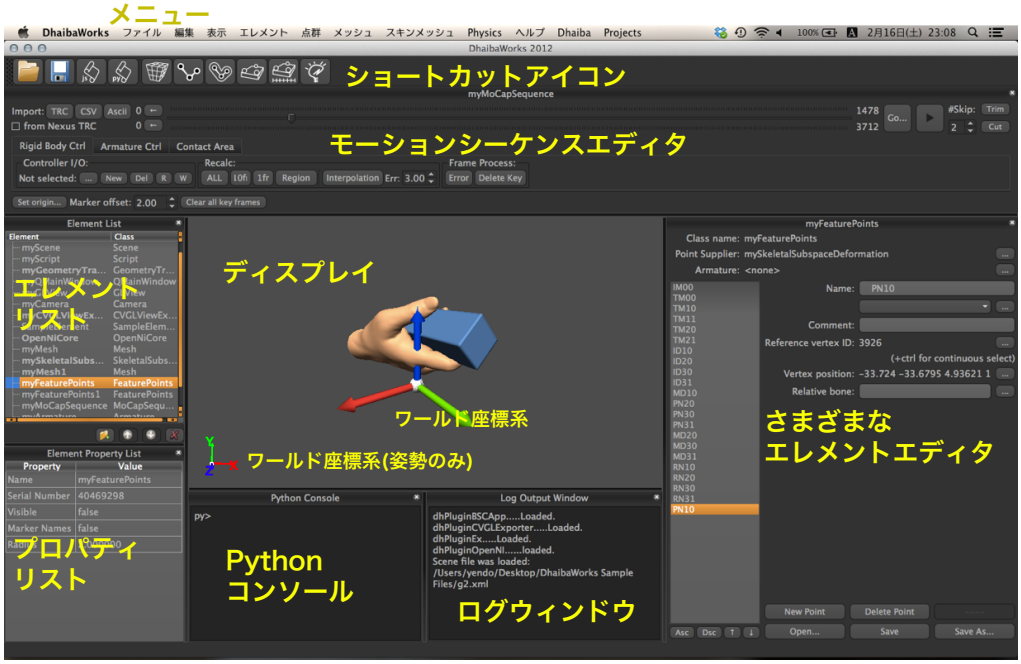
逆動力学解析：操作方法

逆動力学解析の流れ

ここでは、光学式モーションキャプチャを利用して計測された人体動作に対して、DhaibaWorks を利用して逆動力学解析を実行する方法について解説します。その流れは以下の 4 つのステップからなります。

- 人体モデルを生成する
- 人体の動作姿勢を再現する
- 外力を設定する
- 逆動力学解析を実行する

GUI 各部名称

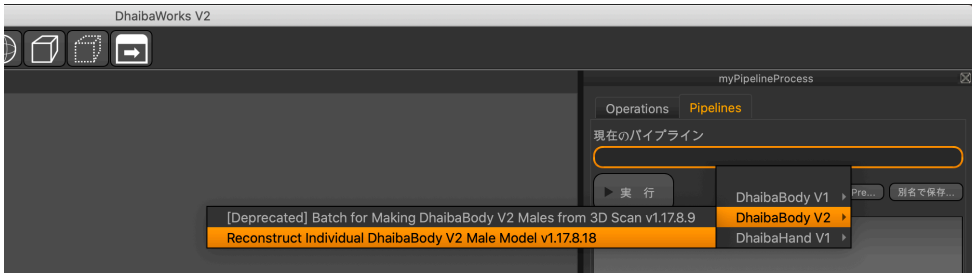


人体モデルを生成する

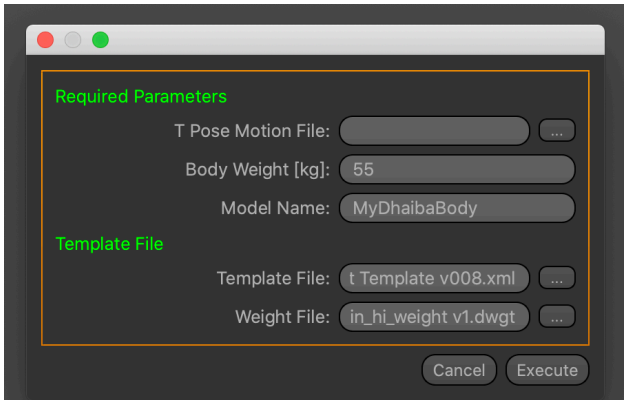
C3D ファイルの初期フレームに立位静止姿勢が含まれる場合

パイプラインエディタから以下の操作を実行する。

- 現在のパイプライン > DhaibaBody V2 > Reconstruct Individual DhaibaBody V2 Male Model v1.17.8.18 を選択する



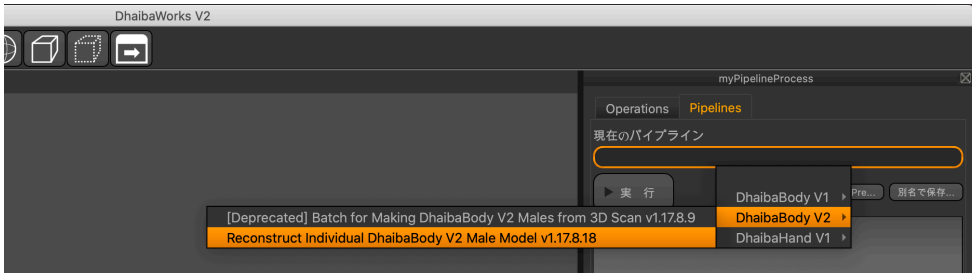
- 「実行」ボタンを押す
- ダイアログが表示される。以下を入力して「Execute」ボタンを押す
  - T Pose Motion File：初期フレームが被験者の立位静止姿勢（T ポーズ）となっている C3D ファイルを選択する
  - Body Weight [kg]：被験者の体重を入力する



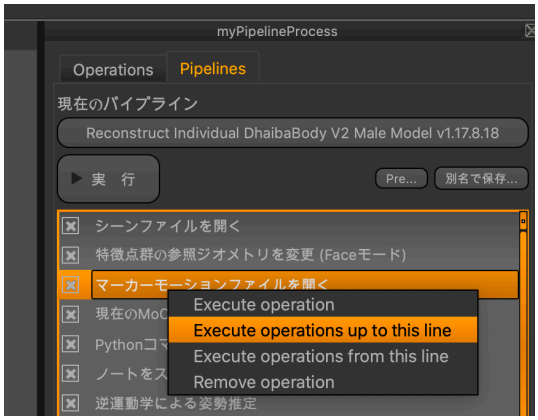
C3D ファイルの途中フレームに立位静止姿勢が含まれる場合

パイプラインエディタから以下の操作を実行する。

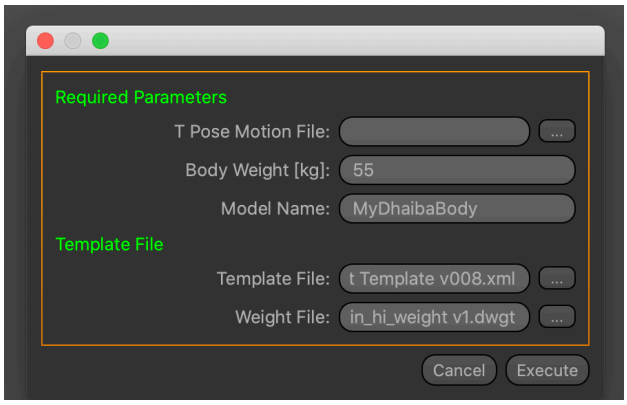
- 現在のパイプライン > DhaibaBody V2 > Reconstruct Individual DhaibaBody V2 Male Model v1.17.8.18 を選択する



- オペレーションエディタ＞マーカーモーションファイルを開く を右クリック、Execute operations up to this line を選択



- ダイアログが表示される。以下を入力して「Execute」ボタンを押す
  - T Pose Motion File：途中フレームが被験者の立位静止姿勢（T ポーズ）となっている C3D ファイルを選択する
  - Body Weight [kg]：被験者の体重を入力する

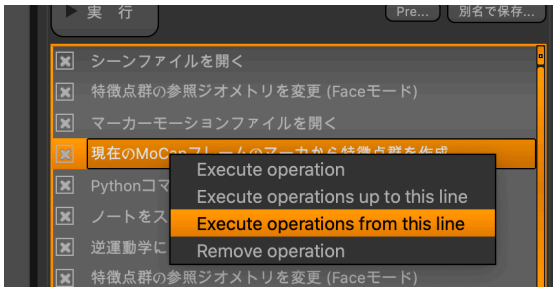


C3D ファイルが読み込まれる。一旦、パイプラインエディタを離れて操作をすすめる。

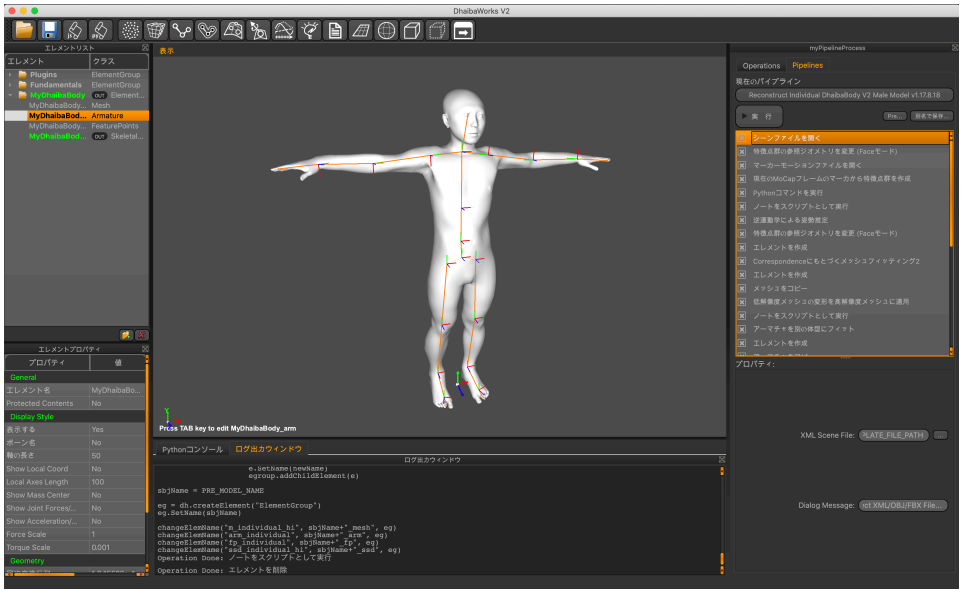
- エレメントリスト＞myMoCapSequence を選択、Tab キーを押して MoCapSequence エディタを開く
- MoCapSequence エディタのスライダーを動かして、被験者の立位静止姿勢（T ポーズ）となっているフレームを表示させる

ふたたび、パイプラインエディタ上で操作をすすめる。

- オペレーションエディタ＞現在の MoCap フレームのマーカから特徴点群を作成を右クリックし「Execute operations from this line」を選択



- 先ほどと同じダイアログが表示されるが、先ほどと同様に入力し「Execute」ボタンを押す
- しばらく待つ。人体モデルが生成される。

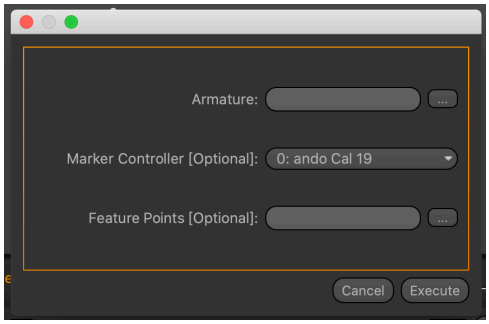


人体の動作姿勢を再現する

- ショートカットアイコン  を押して MoCapSequence エレメントを新規作成する（「myMoCapSequence」エレメントが作成される）

以下、下部に表示された MoCapSequence エディタから操作をおこなう。

- 「Import...」ボタンを押し、再現したい動作を含んでいる C3D ファイルを選択する
- Controllers の「+」ボタン  を押し「dhMCArmatureController」を選択する
- ダイアログが表示される。以下のように入力し「Execute」ボタンを押す
  - Armature :「MyDhaibaBody_arm」を選択する
  - Feature Points :「MyDhaibaBody_fp」を選択する



- Controllers から「MyDhaibaBody_arm」を選択する
- スライダーを動かして、最初のフレームを表示させる
- Landmark Fitting グループから「0」「1fr」「ALL」ボタンを順に押す（「ALL」だけ押しても全フレームの姿勢が再現されるが、最初の方のフレームで再現精度が悪くなることがある）

外力を設定する

逆動力学解析において、人体にかかるすべての外力/トルクをひとつの ForceSet エレメントの中に設定し、MoCapSequence の時刻の変化に追従して、それらの向きや大きさが適切に更新される必要がある。各外力/トルクにおいて設定が必要なものは以下の通り。

- 外力/トルクが人体のどのリンクにかかっているか（時間不変）
- 外力/トルクが現時点で有効かどうか（時間可変）
- 外力/トルクがかかっている位置（時間可変）
- 外力/トルクの向き（時間可変）
- 外力/トルクの大きさ（時間可変）

C3D ファイルに含まれる外力の設定

フォースプレート等から得られた外力データのシーケンスを含んだ C3D ファイルを読み込んだ場合、「myForceSet」エレメントが作成される。その場合、スライダーを動かしてフレームを変更すると、「myForceSet」に含まれる外力/トルクの位置、大きさ、向き、有効性が自動的に変更される。したがって、時間不変の情報のみを設定すればよい。

- エレメントリスト＞myForceSet を選択
- Tab キーを押し、ForceSet エディタを表示させる

以下、ForceSet エディタで操作する。

- アーマチャ(もしくは Armature)の右部にある「...」ボタンを押す
  「MyDhaibaBody_arm」を選択する

- Force リストから各 Force を選択し、以下を設定する
  - ➢ Bone Applied のプルダウンメニューから、この外力がかかっているリンク名を選択する。どのリンクにもかかっていない場合は「ground」を選択する
  - ➢ Inverse Dynamics Parameters＞Identified の「No」をクリックし、「Yes」にする（この外力が既知のものである、という意味）
- Tab キーを押し、ForceSet エディタを閉じる

CSV ファイルに含まれる外力の設定

C3D ファイルとは別に、CSV ファイルから外力データを読み込む場合、時間可変な外力/トルクの情報はすべて CSV ファイルに含めることができる。この場合は、先に

ForceSet エレメントを作成し時間不変な情報を設定しておいたのちに、CSV ファイルを読み込んで時間可変な情報のシーケンスを設定する。

- ショートカットアイコン  から ForceSet エレメントを作成する（「myForceSet」エレメントが作成される）
- エレメントリストから「myForceSet」エレメントを選択する
- Tab キーを押し、ForceSet エディタを表示させる
- 前節「C3D ファイルに含まれる外力の設定」での操作方法と同様に、アーマチャ（もしくは Armature）を設定する
- ForceSet エディタ＞「新規特徴点（もしくは Create...)」ボタンを押し、外力/トルクの個数分、Force オブジェクトを作成する
- Force リストから各 Force を選択し、前節「C3D ファイルに含まれる外力の設定」での操作方法と同様に「Bone Applied」と「Identified」を設定する
- Tab キーを押し、ForceSet エディタを閉じる

- エレメントリストから「myMoCapSequence」を選択する
- Tab キーを押し、MoCapSequence エディタを開く

以下、下部に表示された MoCapSequence エディタから操作をおこなう。

- Controllers の「+」ボタン  を押し「dhMCForceSetController」を選択する。ダイアログが表示されたら「myForceSet」を選択する
- Controllers リスト＞myForceSet を右クリックし「Import ForceSet CSV...」を選択する。ダイアログが表示されたら、外力/トルクのシーケンスを含む CSV ファイルを選択する。CSV ファイルのヘッダ行のフォーマットについては別章を参照のこと

スクリプトを利用した外力/トルクの手動設定

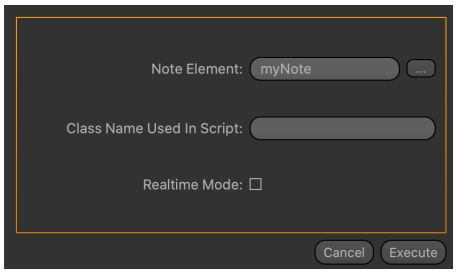
ワイヤ張力など、外力の大きさは分かるものの、その位置や向きは DhaibaWorks 上で幾何学的に計算して設定したい場合などでは、MoCapSequence にスクリプトコントローラを作成し、外力/トルクの時間可変の情報の一部または全部をスクリプトで制御することで、これを実現することができる。

- すでに「myForceSet」エレメントが前節のように何らかの形で作成され、時間不変の情報が設定されているものとする
- ショートカットアイコン  からノートエレメントを作成する（「myNote」エレメントが作成される）
- エレメントリストから「myMoCapSequence」を選択する
- Tab キーを押し、MoCapSequence エディタを開く

以下、下部に表示された MoCapSequence エディタから操作をおこなう。

- Controllers の「+」ボタン  を押し「dhMCScriptController」を選択する
- ダイアログが表示される。以下の通り入力し「Execute」ボタンを押す
  - Note Element :「myNote」を選択する

- Class Name Used In Script :「MyClass」と入力する



- Controllers リスト＞myNote を右クリックし「Copy Script Template to Note」を選択する
  - ノート編集ダイアログが表示される。ここで各 Force を制御するスクリプトを記述する。「OK」ボタンを押すと内容が確定される。再度編集したい場合は、エレメントリスト＞myNote を右クリックし「Edit Note...」を選択する。
  - スクリプトの記述内容については別章を参考のこと。
- Controllers リスト＞myNote を右クリックし「Reset Script」を選択する。
  - myNote のスクリプトの内容を変更した場合は必ずこれを実行する

GUI を利用した外力/トルクの設定

各 Force の位置については、GUI を利用することで、ある特徴点の位置に追従させることもできる。たとえば、あらかじめ作成された特徴点群エレメント「myDhaibaBody_fp」に含まれる左手先の特徴点「LFIN」の位置に Force を配置するには、以下のような操作をおこなう。

- エレメントリストから「myForceSet」を選択する
- Tab キーを押し、ForceSet エディタを開く

以下、ForceSet エディタで操作をおこなう。

- Force リストから、左手先に配置したい Force を選択する
- Reference グループ＞Feature Points から「MyDhaibaBody_fp」を選択
- 同グループ＞Feature Point の「...」ボタンを押し、ディスプレイから左手先の特徴点「LFIN」を右クリックする
  - 「...」ボタンを押すと、選択できる特徴点の色が水色にハイライトされる
- 同グループ＞Position にチェックを入れる

このように、人体モデルに付着している特徴点位置を参照させることで、人体の姿勢が変化しても指定した位置に Force を追従させることができる。ただし、Force の向きと大きさについては、一般的にこの方法では適切に設定できないため、前節までに紹介したような何らかの方法で別途設定しておく必要がある。

逆動力学解析を実行する

逆動力学解析を実行するには、まず、現在の 1 姿勢に対する逆静力学解析を行うための設定を行う。

- メニュー＞エレメント＞Create Element＞Create Multi Rigid Body Force Estimation を選択（「myMultiRigidBodyForceEstimation」エレメントが作成される）
- エレメントリストから「myMultiRigidBodyForceEstimation」を選択する
- エレメントプロパティから以下の項目を設定する（設定するには各プロパティの「値」列をクリックする）
  - Armature :「MyDhaibaBody_arm」を選択
  - Input Force Set :「myForceSet」を選択
  - Gravity：重力加速度ベクトル[m/s^2]を設定する。+Z 軸方向に立っているのであれば [0　0　-9.8　0] のままでよい

以上で逆静力学解析の設定が完了となる。この状態でエレメントプロパティから「Execute」ボタンを押すと、現在の姿勢に対して各リンクの関節トルクを計算する（ひき続き逆動力学解析の設定をおこなう場合は不要)。

- エレメントリストから「myMoCapSequence」を選択する
- Tab キーを押し MoCapSequence エディタを開く

以下、下部に表示された MoCapSequence エディタから操作をおこなう。

- Controllers の「+」ボタン  を押し「dhMCPhysicsRigidBodyController」を選択する
- ダイアログが表示される。「myDhaibaBody_arm」を選択する
- Controllers リストから「MyDhaibaBody_arm_dynamics」を選択する
  - 各フレームにおける各リンクの加速度・角速度等を計算するため、少し時間がかかる
- 「Solve Inverse Dynamics」ボタンを押す。「myMultiRigidBodyForce-Estimation」を選択する
  - 各フレームにおける各関節トルク等を計算するため、少し時間がかかる
- 以上で人体にかかる関節トルクが導出される。Controllers リストから「MyDhaibaBody_arm_dynamics」を右クリックし「Export report...」を選択する。
- ダイアログがあらわれる。以下のように設定し「Execute」ボタンを押す
  - CSV File Path：保存するファイル名
  - Output Dynamics Parameters：関節トルク推定に使用した、各リンクの加速度・角速度等のキャッシュを出力するかどうか
- CSV ファイルとして解析結果が保存される

スクリプトによる外力/トルクの設定

dhMCScriptController 向け Note エレメントの記述

スクリプト（Python）を利用して外力/トルクを設定するためには、前章で述べた通り、MoCapSequence エディタを使用して dhMCScriptController を作成し、これを Note エレメントと紐づける。その後、Controllers リスト＞myNote を右クリックし「Copy Script Template to Note」を選択すると、myNote の内容として、dhMCScriptController 向けのスクリプトテンプレートが挿入される。

```
# In a note element, a class should be defined in Python
# as follows:
# After editing the note, the script should be reset!!

class MyClass:
    def __init__(self):
        # Here member variables are initialized.
        # variables seq and ctrl are automatically set before
        # OnInitialized() is called.
        self.seq = None    # dhMoCapSequence element
        self.ctrl = None   # dhMCScriptController object

    def OnInitialized(self):
        # This method is called in Controller->Initialize().
        pass

    def OnPropertyUpdated(self, globalFrameId):
        # This method is called in Controller->updateProperty()
        pass
```

■ テンプレートを挿入した状態での myNote

dhMCScriptController オブジェクトが作成またはリセットされると、同時に myNote で定義した MyClass オブジェクトが作成され、OnInitialized()関数が呼ばれる。また、MoCapSequence エレメントが再生中の際は、フレームが更新されるごとに OnPropertyUpdated()関数が呼ばれる。その際、globalFarmeId には現在のフレーム番号が設定されている。

```
import math

class MyClass:
    def __init__(self):
        self.seq = None    # dhMoCapSequence element
        self.ctrl = None   # dhMCScriptController object
        self.mass = 0.0
        self.markers = None
        self.forces = None
        self.fplateL = None
        self.fplateR = None
        self.fhandL = None
        self.fhandR = None

    def markerPosition(self, markerName):
        return self.markers.point(markerName).position()

    def markerPositions(self, markerNames):
        positions = []
        for markerName in markerNames:
            pos = self.markers.point(markerName).position()
            positions.append(pos)
        return positions

    def OnInitialized(self):
        self.markers = dh.element("Markers_fp")
        self.forces = dh.element("myForceSet")
        self.fplateL = self.forces.force("Force1")
        self.fplateR = self.forces.force("Force0")
        self.fhandL = self.forces.force("HandL")
        self.fhandR = self.forces.force("HandR")

        self.seq.goToFrameAt(1188)

        fsum = self.fplateL.force() + self.fplateR.force()
        self.mass = fsum.value(2)

    def OnPropertyUpdated(self, globalFrameId):
        fsum = self.fplateL.force() + self.fplateR.force()
        currentMass = fsum.value(2)
        if currentMass < self.mass:
            self.fhandL.setEnabled(False)
            self.fhandR.setEnabled(False)
        else:
            vf = 0.5 * dhVec4(0, 0, self.mass - currentMass, 0)
            self.fhandL.setForce(vf)
            self.fhandR.setForce(vf)
            self.fhandL.setEnabled(True)
            self.fhandR.setEnabled(True)
```

■ myNote のスクリプトサンプル

DhaibaWorks V2 Python API リファレンス（抜粋）

スタティック関数

```
Element dh.element(ElementName)
```

名前が *ElementName* (str 型)の Element オブジェクトを返す

```
Vec4 dhVec4(X, Y, Z, W)
```

$[X, Y, Z, W]^T$ のベクトルを作成する

ForceSet エレメントのメンバ関数

```
Force force(ForceName)
```

名前が *ForceName* (str 型)の Force オブジェクトを返す

Force オブジェクトのメンバ関数

```
Vec4 force()

Vec4 torque()

Vec4 position()
```

Force オブジェクトの力ベクトル/トルクベクトル/位置ベクトルを返す

```
bool isEnabled()
```

Force オブジェクトが有効かどうかを返す

```
setForce(Value)

setTorque(Value)

setPosition(Value)
```

Force オブジェクトの力ベクトル/トルクベクトル/位置ベクトルを *Value* (Vec4 型)とする

MoCapSequence エレメントのメンバ関数

```
goToFrameAt(GlobalFrameId)
```

グローバルフレーム番号 *GlobalFrameId* (int 型)のフレームに移動する

FeaturePoints エレメントのメンバ関数

```
FeaturePoint point(PointName)
```

名前が *PointName* (str 型)の FeaturePoint オブジェクトを返す

FeaturePoint オブジェクトのメンバ関数

```
Vec4 position()
```

FeaturePoint オブジェクトの位置ベクトルを返す

Vec4 オブジェクトのメンバ関数

```
float value(Index)
```

4 次元ベクトルの *Index* 番目の要素を返す

外力/トルク CSV ファイルのフォーマットについて

外力/トルク CSV ファイルについて

前章までに述べたとおり、MoCapSequence エディタにおいて dhMCForceSetController を作成し、Controllers リストからこのコントローラを右クリックし「Import ForceSet CSV...」を選択することで、特別なフォーマットで記載された CSV ファイルを、外力/トルクのシーケンスデータとしてインポートすることができる。サンプルは以下の通り（Excel にて表示）。

	A	B	C	D	E	F	G	H	I
1	FrameId	Force1.Position.X	Force1.Position.Y	Force1.Position.Z	Force1.Force.X	Force1.Force.Y	Force1.Force.Z	Force1.Enabled	Force2.Force.Z
2	1	100	200	300	10	20	100	1	1000
3	2	200	200	300	10	20	200	1	900
4	3	300	200	300	10	20	300	1	800
5	4	400	200	300	10	20	300	0	700
6	5	400	200	300	10	20	300	0	700
7	6	400	200	300	10	20	300	0	700
8	7	400	200	300	10	20	300	0	700
9	8	400	200	300	10	20	300	0	700
10	9	400	200	300	10	20	300	0	700
11	10	400	200	300	10	20	300	0	700
12									

第一行はヘッダ行、第二行以降には、各時刻における各 Force の値が入る。第一列には「FrameId」が入る。これは、計測された周波数に対応したフレーム番号である。例えば、100Hz で計測されたデータシーケンスにおいてフレーム番号が 100 のとき、計測開始から 1 秒後のデータであることを示す。

第二列以降には各フォースにおける情報が入る。例えば、「Force1.Position.X」であれば、「Force1」という名前の Force の位置について、その X 座標の値が入る。前例における「Position」の部分には、下記の項目のいずれかが入る

- Position：Force の位置
- Force：Force の力ベクトル
- Torque：Force のトルクベクトル
- Enabled：その Force がその時刻において有効かどうか（0: 無効、1: 有効）
  - ➢ Enabled の場合は後ろに「.X」などの座標名を付記しない