

## ▼ Home

自己紹介  
実績  
マラソン記録  
リンク

## ▼ Welcome to my home page (English)

Career  
Publication

## ▼ OpenSim Tutorial

## ▼ 入門編

1. チュートリアル I - 筋骨格モデル入門
2. チュートリアル II - 腰移行手術のシミュレーションと解析
3. チュートリアル III - スケーリング、逆運動学、逆動力学
4. サッカーキック
5. 足関節損傷予防シミュレーション
6. 運動代謝コスト
7. ダイナミックウォーキングチャレンジ
8. 静的最適化

▶ 中級編

▶ 上級編

サイトマップ

## 6 . 運動代謝コスト

このページは [Simulation-Based Design to Reduce Metabolic Cost](#) に対応しています

[Metabolics Example Files](#) をダウンロード

### 概要

この項では歩行を補助する装置を作成し、その評価のためオープンシムソフトウェアをどのようにして用いるかを説明します。ここでは簡素化した筋骨格モデルを用いて歩行を解析します。モデルの被験者は体重75kg、身長1.8m、速度1.2m/sでトレッドミル上を歩行しています。静的および動的歩行装具を用いて筋駆動モデルシミュレーションの方法を学び、歩行中のエネルギー代謝への影響を調べます。オプションで代謝計算プローブをモデルに適用する方法や、装具をモデルに追加するスクリプトシェルのオープンシムAPIライブラリーを使う方法も学べます。

### I. モデルを調べる

初めに筋骨格モデルの構成を調べます。この後でオープンシムを起動させ、Navigator パネルでbodies, joints, musclesを確認してもらいます。歩行装具を取り付けて何度かシミュレーションを行うので、シミュレーションの実行速度が速くなるようモデルを簡素化しています。

特徴：

- 頭部、上肢、体幹はtorso(胴体)セグメント1つで表わされています。
- 股関節、足関節、体幹と骨盤の腰部関節はすべて1自由度です。
- 下肢筋の数を減らしており、モデルは屈曲及び伸展の主要筋のみモデル化しています。

また、今回は筋駆動シミュレーションと代謝コストに着目しているため、その前処理に必要なスケーリングとRRA (Residual Reduction Algorithm) が行われたモデルから始めます。

\* RRAはこの後のチュートリアル「静的最適化」で説明します。

特徴：

- Scaleツールにより被験者の体格（体重、セグメント長など）に合わせられたモデルを使います。
- 逆運動学ツールを用いて測定したマーカー軌跡を追うようにモデル運動が調整されています。
- 測定した床反力でモデルが測定動作を追従できるよう、セグメントの関節モーメントと関節運動の調整をするResidual Reduction Algorithmツールが使われています。

詳しく知りたい方はユーザーズガイドで [Scaling](#), [Inverse Kinematics](#), [Residual Reduction Algorithm](#) の項目を読んでください。

## A. オープンシムの起動

ウィンドーメニューからオープンシムを起動してください。複数のパネルが開きます。Viewパネルにモデルが表示されている場合はモデルを閉じてください。

## B. モデルの読み込み

1. オープンシムGUIで**File>Open Model....**を選択してください。
2. **Models\Gait10dof18musc** (または**gait10dof18musc\_32**, \*ここではgait10dof18musc\_32に統一します) のフォルダに移動してください。
3. **subject01\_metabolics.osim**を開いてください。このモデルはRRA実施後のモデルです。オープンシムGUIは骨格モデルが表示されていることを確認してください。
4. **Navigator**パネルで**+ アイコン**によりモデルの構成要素を開いてください。これでモデルのbodies, joints, forcesなどが確認できます。
5. モデルの関節角度を表示するので、**Coordinates**をクリックして下さい。スライダーで角度の調整ができます。
6. **View**パネルで足関節を拡大してみましょう。マウスの右ボタンでクリックとドラッグすると拡大、縮小ができます。真ん中のボタン (右左ボタンの両方) で平行移動、左ボタンで視野を回転できます。
7. 読み込んだRRAモデルと元のモデルを比較します。
  - **Coordinates**パネルで**Poses>Default**をクリックし、モデルを初期姿勢に戻してください。モデルをズームアウトして全身が見える状態にしてください。Viewパネルの右端にあるボックスアイコンを使ってX, Y, Z軸から見ることもできます。
  - 元の初期モデルを開きます。**File>Open Model...**で**gait10dof18musc\_32**フォルダから**gait10dof18musc.osim**を選択してください。読み込んだモデルがNavigatorパネルで太字になり、カレントモデルとして表示されます。ViewウィンドーでRRAモデル (**subject01\_metabolics.osim**) の表示は透明になっています。
  - 二つのモデルの体幹セグメント質量を比較します。Navigatorパネルでbodyのtorsoをクリックしてハイライトにすると、画面上左下のPropertyエディターに質量が表示されます。
  - viewウィンドーでモデルサイズを比較してください。
  - すべて終わったら、Navigatorパネルで基本モデルの名前を右クリックし、ドロップダウンメニューからCloseを押してください。

## Questions

1. モデルの関節の自由度、筋、セグメントの数を調べてください。
2. 腰部の関節をまたぐ筋を調べてください? **Navigator**パネルで筋をハイライトにし、左下の**Properties**ウィンドーで**Geometry Path**の横の...ボタンをクリックしてください。筋は歩行中の体幹運動をコントロールするのに十分な力を持っていますか?
3. BMI(body mass index)が低いのはどちらのモデル (初期モデルか被験者の体格に合わせたモデル) ですか? BMI = 体重 [kg] / 身長 [m]<sup>2</sup>

Questionが終わればNavigatorウィンドーのgait10dof18musc.osimモデルを右クリックし、Closeを選択してモデルを閉じてください。

## II. 通常歩行のシミュレーション

このセクションではオープンシムのComputed Muscle Control (CMC)ツールを用いて通常歩行のシミュレーションを行います。上で記載したように、RRAツールによって計算された運動力学データを元に進めます。CMCのアルゴリズムは入力された関節運動を (error tolerance以下の値で) 追従するように筋の興奮度を決定します。CMCは筋活動の二乗和を最小にすることで筋の冗長性 (関節運動よりも筋の数が多いために解が複数存在すること) を解決します。CMCの詳細はユーザーズガイドの [Computed Muscle Control](#) を参考にしてください。シミュレーションが終わった後に、元のデータと出力結果がどの程度一致するかを確認します。

## A. 動作の読み込み

1. カレントモデルがwalk\_subject01であることを確認してください。File>Load Motion...を選択し、gait10dof18musc\_32\RRA\ResultsRRAとフォルダを移動してsubject\_adjusted\_Kinematics\_q.stoを選択してください。(Models\Gait10dof18muscフォルダにはResultsRRAが含まれていません。)
2. Motion Sliderパネル (画面中央上の青色矢印) で動作を再生します。Motion sliderパネルから再生、停止、フレーム送り、逆再生、フレーム逆送りができます。スライダをスクロールすることでも動かすことができます。
3. 動作中の床反力を表示することもできます。NavigatorパネルからMotionsのCoordinates を右クリックし、Associate Motion Data...を選ぶとフォルダが開きます。gait10dof18musc\_32\ExperimentalData\subject01\_walk\_grf.motを選択し、Openを押してください。動作を再生すると計測データに応じた床反力の大きさとベクトルが緑の矢印で表示されます。
4. 歩行周期の特定のタイミングを調べることができます。歩行周期は一側下肢の踵接地で規定できます。右足の踵接地を表示し、歩行周期開始タイミングを調べてください。

## B. 筋駆動シミュレーションComputed Muscle Control (CMC)

- Tools>Computed Muscle Control を選択してCMCツールを開いてください。CMCパネルでシミュレーションの設定をします。このツールでGUIのカレントモデル(今回はwalk\_subject01)を操作します。walk\_subject01を選択してカレントモデルに変更してください。
- CMCの設定は準備されているセッティングファイルを用います。Load...を押して、CMC\walk\_Setup\_CMC.xmlを選択してください。ファイルはCMCフォルダにあります。
- Main SettingsでInputの枠を見てください。
  - Desired kinematicsではセクションAでも読み込んだRRA計算後の運動学データ(subject\_adjusted\_Kinematics\_q.sto)が選択されています。
  - filter kinematicsにチェックを入れないでください。このデータはRRAの際にすでに平滑化処理がされています。
  - Tracking tasksファイルgait10dof\_Kinematics\_Tracking\_Tasks.xmlはオープンシムのトラッキングする項目 (今回はすべての関節運動) とセグメント質量を指定します (右横の鉛筆ボタンを押すと、トラッキングファイルの詳細を確認できます)。
  - Time rage to processで解析する歩行区間を決めます。セクションAで調べた値が入っています (右踵接地から次の右踵接地まではおおよそ0.6から1.9秒です)。
- 実行するにはresidual アクチュエーターとreserveアクチュエーターの設定および床反力データが必要になります。これらはActuators and External Loadsタブで確認できます。
  - Additional force set fileにgait10dof\_Reserve\_Actuators.xmlが表示されています。このファイルはモデルの関節に作用するアクチュエーターと骨盤に作用するアクチュエーターの設定が入力されています。これらのアクチュエーターは計測動作や床反力データとモデル動作が力学的に一致してシミュレーションが動くように、関節や骨盤に必要な力を補います。
  - "Append to model's force set"ボタンが選択されています。これはモデルで組み込まれている筋にreserveアクチュエーターやresidualアクチュエーターを追加して解析することを意味しています。
  - また、External loads specification fileにExperimentalData\subject01\_walk\_grf.xmlが設定されています。これはモデルの足セグメント (calcn\_rとcalcn\_l)にsubject01\_walk\_grf.motファイルの値を加える設定になっています。右横の鉛筆ボタンをクリックすると、設定の詳細を確認できます。

- **Run**をクリックしてください。**Close**を押してCMCツールを閉じてください。CMC計算は進んでモデルが動き続けます。

### C. シミュレーション結果の評価

- **Motion Slider**パネルで結果を**再生**してください。青から赤に変化している筋が活動していることを表します。
- 次はCMCシミュレーションが入力した運動と一致するかを調べます。
  - **Tools>Plot...**を選択し、Plotterウィンドーを開いてください。
  - **Y-Quantity...**を選び、一番下にある**Load file...**を選択してください。CMC結果が保存されているフォルダ(**CMC\ResultsCMC**)から**walk\_subject\_Kinematics\_q.sto**のファイルをクリックしてください。**pelvis\_tilt, hip\_flexion\_r, knee\_angle\_r, ankle\_angle\_r**のすべてのチェックボックスをオンにして**OK**を押してください。
  - **X-Quantity...**を選び、**time**を選択します。**Add**ボタンをクリックして、曲線を描きましょう。
  - これで運動学データが読み込まれました。**RRA\ResultsRRA**フォルダ内の**subject\_adjusted\_Kinematics\_q.sto**を使ってbとcのステップを同じように繰り返しましょう。
  - 運動学データの一致度を調べます。カーソルを曲線上に動かしてください。調べたい曲線の近くを拡大できます。クリックとドラックを使って調べたい領域を左上から右下に囲い、拡大してください。拡大を元に戻したい時はクリックとドラックを使って反対方向（右上から左下など）に囲ってください。
  - この後の操作で使うプロットを使うため、最小化にして閉じないでください。
- 上記手順でプロットを作りましたが、**scripts**を使って作成することもできます。CMC結果を調べるプロットスクリプトが用意されています。このスクリプトは結果を簡単に表示でき便利です。スクリプトを実行しましょう：
  - **Scripts>Run...**を選択してください。
  - **gait10dof18musc\_32\Scripts**のフォルダを探し、**CMC\_EvaluateResults.py**を選択してください。
  - スクリプトを始めるにはフォルダを選択する必要があります。CMC結果を保存するフォルダ**CMC\ResulCMC**を選択してください。
  - スクリプトから3つのプロットを調べます。
    - 運動学データの違い。すでにプロットしたRRAとCMC曲線の差です。歩行測定でこの差は普通は2度（0.035ラジアン）未満になります。
    - residual forceとmomentが骨盤に加えられています（「神の手の力」\* 順動力学計算を数学的に成り立たせるために加えられた仮想の力）
    - reserveアクチュエータが出力するトルクが関節に生じています。\*scriptの詳細を確認したい時は**Scripts>Open...**から**CMC\_EvaluateResults.py**を選択してください。確認すれば閉じてください。
  - plotウィンドーを最小化にしてください。アシスト歩行のシミュレーションで再度使います。
- 最後にシミュレーションの筋張力を調べます。今回は足関節筋を見ましょう。
  - **Tools>Plot...**により別のPlotterウィンドーを開きます。
  - **Y-Quantity...**の**Load file...**からCMC resultsフォルダ内の**walk\_subject\_Actuation\_force.sto**を選択してください。
  - 右下肢の腓腹筋とヒラメ筋、前脛骨筋(**gastroc\_r, soleus\_r, and tib\_ant\_r**)のチェックボックスをオンにして、**OK**を押してください。
  - **X-Quantity...**から**time**を選択してください。
  - **Add**ボタンを押して曲線を描きます。Curves Listの**Properties...**をクリックし、プロットタイトルをMuscle Forcesに変更してください。

### Questions

1. どの角度データでCMCとRRAの差が最も大きいですか？最大エラー値を調べましょう。角度データは2度（0.035ラジアン）未満に収まっていますか？
2. Residual forceの最大値を調べてください。Residual forceが動作を通じて10N以下の場合、歩行シミュレーション結果が“Good”としています。また、25N以下なら“OK”と規定します（User's guideの[Best practice](#)を参照）。なぜresidualがOKの状態なのでしょう？（Residual momentは50Nm以下であれば“Good”とみなします）
3. なぜ腰椎伸展のreserve actuatorは股関節や膝関節、足関節よりも大きな値になるのでしょうか？
4. 足底屈筋の力発揮ピークタイミングを確認してください。足背屈筋はどうでしょう？

Residual forceが十分に小さな値であることを確認するにはどうすればよいのでしょうか？今回紹介する値は大まかな値であり、一つのガイドラインとみなしてください。Residual forceのピークパワーは4ワットよりも低いですが、足関節筋と腸腰筋が発揮するパワーは歩行時のピークが120ワットを超えます。このため、residualは全体のエネルギーから考えるとごくわずかであると言えます。さらに、CMCを実行するタイムウィンドー（CMC計算のtime interval、[How CMC Works](#)を参照）を小さくすれば、関節運動の誤差を減らし、スムーズなパワーカーブを描くことができます。研究で用いる際に精度高く解析する場合はシミュレーションステップの数を増やすとよいでしょう。

### III. 通常歩行の運動代謝解析

筋駆動モデルシミュレーションを使って歩行動作の代謝コストを調べることができます。オープンシムでの代謝計算ではモデルの筋と体のエネルギー消費総量を算出します。Umbergerらの2003年2010年の論文を元に、モデルでは筋質量、速筋・遅筋線維の割合、伸長・短縮速度の影響を考慮しています。セクションIとIIで動かしたモデルには筋のエネルギーコストを計算する代謝プローブが用意されています。代謝プローブに関しては[doxygen](#)のページを参照してください。

#### A. 代謝プローブ

1. **Navigator**パネルでモデルの**Probes**セットを開き、metabolicsをクリックしてください。
2. スクリーンの左下にある**Propertyエディター**でモデルの代謝プローブを調べます（またはWindow>Properties）。
  - **probe\_operation**を見つけてください。右の...ボタンを押して'value'を'integrate'に書き換えてみてください。何が変わりましたか？initial\_conditions\_for\_integrationは0に設定されましたか？
  - **aerobic\_factor**を見つけてください。歩行動作は有酸素運動か無酸素運動どちらだと思いますか？また、歩行に適した値が入力されていますか？（aerobic\_factorをクリックすると下に説明が表示されるので、参考にしてください。）

#### B. 歩行シミュレーションの代謝計算

1. 次に結果を解析します。CMCを行った時に歩行周期の各筋の代謝とすべての筋のエネルギー消費量を計算したProbeReporter解析が同時に動いています。
2. **Tools>Plot...**を選択してPlotterウィンドーを開いてください。
3. **Y-Quantity...**をクリックし、**Load File...**よりCMC結果を保存したフォルダ（CMC\ResultsCMC）で**walk\_subject\_MetabolicsReporter\_probes.sto**ファイルを開いてください。**metabolics\_TOTAL**のチェックボックスをオンにして**OK**を押してください。
4. **X-Quantity...**で**time**を選択してください。

5. **Add**をクリックすれば、曲線が表示されます。CMC解析時にprobe\_operationで'value'が選択されていれば、歩行中の各タイミングの代謝エネルギー消費（単位はワット）が計算されます。
6. 右の足部底屈筋(gastroc\_r and soleus\_r)のエネルギー代謝を同様に表示してください。

## Questions

1. 歩行中にエネルギーコストが最も高くなるのはどのタイミングですか？
2. セクション II.Cで行った筋発揮張力のプロットを見てください。ヒラメ筋は0.9秒、1.3秒でピーク値を示しました。ヒラメ筋のエネルギー代謝は立脚後期よりも立脚中期の方が低くなっているはずですが、力発揮と代謝コストの違いはなぜ生じていると思いますか？（初めのヒラメ筋のピークは歩行中に普段計測されません。このチュートリアルでは単純なモデルを使っていることを思い出してください）
3. 追加の質問：一歩行周期の消費エネルギーは何ジュールですか？歩行全体の平均コストは何J/kg/s (または W/kg)ですか？1.2m/sの歩行では4-6W/kgと報告されていますが、この値と比較して高いでしょうか低いでしょうか？ probe\_operationを'integrate'にして再度CMC解析を行ってみましょう。

## IV. モデルの作成

このセクションではscripting shellとProperty Editorを用いて右下肢に簡単な装具を取り付けた二つのモデルを作成します。今回は簡単なモデルを作成しますが、オープンシムソフトウェアでは複雑なアクチュエーターやコントローラに発展させることも可能です。gait10dof18muscleフォルダには2つのモデルが準備されており、このセクションを飛ばしてセクションVに進むこともできます。モデル作成のためscripting shellとProperty Editorを使いましょう：

Note: scripting shellでcommand prompt (>>>)にコードを張り付けることはできません。GUIで新しいモデルを作り、実行するコードを保存するには新しいpyファイルを作ることをお勧めします。オープンシム3.2バージョン（3.3も同様？）ではコードを間違えて入力した時はエスケープキーを押すと止めることができます。

### A. 足部スプリングモデルの作成

足関節に回転スプリングモデルCoordinateLimitForceを作ります。装具が発する力はスプリングやダンパーによって生じ、設定した関節角度を超えると関節運動を制限します。足関節の角度が設定角度を超えた時に、足関節にスプリングとダンパー用の力が働くCoordinateLimitForceを用います。CoordinateLimitForceは足関節背屈角度が5度を超えると、ステイフネス値K\_upper (10 N-m/degree)に応じて底屈モーメントを生み出す力を発生させます。Transitionは装具の硬さが0（リミット前）からK（リミット以上）に移行する区間（関節角度）を設定しています。

subject01\_metabolics.osim m

odeが読み込まれており、カレントモデル（Navigatorパネルでwalk\_subject01が太字となっている）になっていることを確認してください。ScriptingShellウィンドウ（Viewウィンドウの下）でコマンドプロンプト >>>を探してください。ScriptingShellウィンドウが見つからない場合はWidowsからScriptingShellを選択してください。回転スプリングをモデルに追加して新しモデルを作成するには下記のコマンドを入力します。（コード全体をコピー＆ペーストはできません。一行ずつコピー＆ペースト後にエンターを押すことでコードを張り付けられます。下図を参照してください。）：

```
Messages ScriptingShell Window x
Jython Interactive Console
>>> # Get a handle to the current model and create a new copy
>>> baseModel = getCurrentModel()
>>> ankleSpringModel = baseModel.clone()
>>> ankleSpringModel.setName(baseModel.getName() + '_ankle_
>>> # Create the spring we'll add to the model (a CoordinateLi
>>> ankleSpring = modeling.CoordinateLimitForce()
>>> ankleSpring.setName('AnkleLimitSpringDamper')
>>> # Set the coordinate for the spring
>>> ankleSpring.set_coordinate('ankle_angle_r')
>>> # Add the spring to the model
>>> ankleSpringModel.addForce(ankleSpring)
>>> # Load the model in the GUI
>>> loadModel(ankleSpringModel)
>>>

Command History
## Get a handle to the current model and create a new copy
## baseModel = getCurrentModel()
## ankleSpringModel = baseModel.clone()
## ankleSpringModel.setName(baseModel.getName() + '_ankle_spring')
## Create the spring we'll add to the model (a CoordinateLimitForce in OpenSim)
## ankleSpring = modeling.CoordinateLimitForce()
## ankleSpring.setName('AnkleLimitSpringDamper')
## Set the coordinate for the spring
## ankleSpring.set_coordinate('ankle_angle_r')
## Add the spring to the model
## ankleSpringModel.addForce(ankleSpring)
## Load the model in the GUI
## loadModel(ankleSpringModel)
```

```
#Get a handle to the current model and create a new copy
baseModel = getCurrentModel()
ankleSpringModel = baseModel.clone()
ankleSpringModel.setName(baseModel.getName() + '_ankle_spring')

# Create the spring we'll add to the model (a CoordinateLimitForce in OpenSim)
ankleSpring = modeling.CoordinateLimitForce()
ankleSpring.setName('AnkleLimitSpringDamper')
# Set the coordinate for the spring
ankleSpring.set_coordinate('ankle_angle_r')

# Add the spring to the model
ankleSpringModel.addForce(ankleSpring)

# Load the model in the GUI
loadModel(ankleSpringModel)
```

これでモデルにスプリングが追加されました。GUIのPropertyエディターで値を設定します。

1. **Navigator**の**walk\_subject01\_ankle\_spring**で**Forces**から**Other Forces**を探してください。
2. **AnkleLimitSpringDamper**をクリックし、**Properties**ウィンドウで値を設定します。
  - upper\_stiffness = 10.0 足関節背屈をアシストするスプリングの硬さ
  - upper\_limit = 5.0 足関節背屈5度以上でスプリングが作用
  - lower\_stiffness = 1.0 CoordinateLimitForceが発する底屈制動力はlower\_stiffnessとlower\_limitで規定
  - lower\_limit = -90.0 足部運動はすべての底屈角度で運動可能

- damping = 0.01 小さなダンパー力
  - transition = 2.0 ここでは関節角度がlower\_limitかupper\_limitを超えた時に装具の硬さが0から一定の硬さに移り変わるまでの角度範囲(°)を規定
3. 新たにモデルを保存します。Navigatorパネルのモデル名 (walk\_subject01\_ankle\_spring) で右クリックを押してSave As...を選択してください。保存名はsubject01\_metabolics\_spring.osimとします。(作り上げたモデルはすでにフォルダ内に用意されています。よって上書き保存をする必要はありません。)

GUIでモデルを読み込んだ後に新しい要素を追加することはできません。今ある要素のPropertiesのみ変更するが可能です。このため、ScriptingShellで入力したankleSpringModel = baseModel.clone()でクローンモデルを使い、モデルに新たな要素を追加しています。

## B. 二つの関節をまたぐスプリングモデルの作成

次は大腿から足部セグメントに走行する静的スプリングをモデルに加えます。スプリングの走行は筋と同様にGeometryPathで変更できます。腓腹筋のステイフネスを10000N/mに増加してスプリングを作成しましょう。セクションI, II, IIIで読み込んだsubject01\_metabolics.osim (Navigatorパネルではwalk\_subject01) がカレントモデルになっているか再度確認してください。ScriptingShellのマンドプロンプト >>>を探してください。下記のコマンドを入力して回転スプリングを作り上げ、新しいモデルを作成します。GUIのProperty エディターから値を入力する代わりにスプリングの設定を入力することにします。

```
# Get a handle to the current model and create a new copy
baseModel = getCurrentModel()
pathSpringModel = baseModel.clone()
pathSpringModel.setName(baseModel.getName() + '_path_spring')

# Create the spring we'll add to the model (a PathSpring in OpenSim)
name = 'BiarticularSpringDamper'
restLength = 0.4
stiffness = 10000.0
dissipation = 0.01
pathSpring = modeling.PathSpring(name,restLength,stiffness,dissipation)

# Set geometry path for the path spring to match the gastrocnemius muscle
gastroc = pathSpringModel.getMuscles().get('gastroc_r')
pathSpring.set_GeometryPath(gastroc.getGeometryPath())

# Add the spring to the model
pathSpringModel.addForce(pathSpring)

# Load the model in the GUI
loadModel(pathSpringModel)

# Save the model to a file
fullPathName = baseModel.getInputFileName()
newName = fullPathName.replace('.osim', '_path_spring.osim')
pathSpringModel.print(newName)
```

これらのコマンドはgait10dof18muscle\ScriptsフォルダにaddAnkleSpring.pyとaddPathSpring.pyとしてファイルが保存されています。

## V. 静的装具を使った歩行シミュレーション

セクションIIIで通常歩行のデータは解析できています。補助装具を使って代謝コストが減少するか調べてみましょう。足関節のみに作用する装具と、腓腹筋のように2関節に作用する装具の2つを試します。gait10dof18muscleフォルダ内にモデルが準備されています。

### A. 足部スプリングを有する筋駆動シミュレーションの実行と結果の確認

1. 足関節スプリングモデル(subject01\_metabolics\_spring.osim)を開いてカレントモデルにしてください。
2. モデルの**Force**リストでankle springを探してください。スプリングのプロパティを見てください。セクションIVで足部スプリングモデルは詳しく説明されています。
3. CMCツールを使い、**CMC**フォルダからwalk\_Setup\_CMC.xmlのセッティングファイルを読み込みましょう。
4. 結果を出力するフォルダを変更するためOutputのDirectoryを**ResultsCMC\_AnkleSpring**に変更してください。他の設定は変更せずにCMC解析を始めます。
5. **Run**をクリックした後、**Close**を押してCMC Toolウィンドウを閉じてください。
6. 装具なしのモデルとの結果を比較します。CMCの解析が終了すればscriptsからRun...を選んで、**plotBaselineResults.py** (Scriptsフォルダにあります)を選択しましょう。保存フォルダはCMC\ResultsCMCとします。このスクリプトは1歩行歩行周期の消費代謝エネルギーの合計と腓腹筋、ヒラメ筋、前脛骨筋、腸腰筋の各筋の代謝量を表示させます。Note: これらのプロットは次のデータと比較するので閉じずに、最小化にしてください。
7. 次に足部スプリングを追加したシミュレーション結果を追加します。**addPlotDeviceResults.py**のスクリプトを選択してください。保存ファイルは先ほどと同様にResultsCMC\_AnkleSpringとします。このスクリプトは前に作成したプロットに足部スプリングモデルの結果を追加します。
8. 次のセクションでも同じプロットを使うので、プロットを再度最小化にしてください。

### B. 二関節スプリングモデルの筋駆動シミュレーションの実行と結果の確認

1. 足関節スプリングモデル(subject01\_metabolics\_path\_spring.osim)を開いてカレントモデルにしてください。
2. モデルの**Force**リストでpath springを探してください。スプリングのプロパティを見てください。Viewパネルでは緑色でスプリングが表示されています。スプリングの走行はどの筋と同じですか？セクションIVで詳しく説明されています。
3. CMCツールを使い、**CMC**フォルダからwalk\_Setup\_CMC.xmlのセッティングファイルを読み込みましょう。
4. 結果を出力するフォルダを変更するためOutputのDirectoryを**ResultsCMC\_PathSpring**に変更してください。他の設定は変更せずにCMC解析を始めます。

5. **Run**をクリックした後、**Close**を押してCMC Toolウィンドウを閉じてください。
6. 二関節スプリングモデルの結果を追加します。CMCの解析が終了すればscriptsから再度**addPlotDeviceResults.py**を選択しましょう。**Run Recent>**を使うとPartAで用いたスクリプトが実行できます。保存フォルダは**ResultsCMC\_PathSpring**とします。このスクリプトは先ほどのプロットに二関節スプリングモデルの結果を追加します。

## Questions

1. Total Metabolic Energyを見てください。どの装具で代謝コストが減少していますか？
2. 右の腓腹筋、ヒラメ筋、前脛骨筋の立脚初期から中期の代謝量を調べてみましょう。装具なしと比較して装具では足部底屈筋のエネルギー消費が増加していますか？減少していますか？装具は前脛骨筋にどのような影響を及ぼしましたか？なぜこのような変化が生じると思いますか？
3. 立脚後期と遊脚期のヒラメ筋と腸腰筋の代謝を見てください。二関節スプリング結果はどのように変わっていますか？
4. 作成したスプリングがトラッキングエラーやresidual, reserve forceに影響を与えていないかを調べます。**CMC\_EvaluateResults.py**スクリプトを2回実行します。はじめは**ResultsCMC\_AnkleSpring**、次に**ResultsCMC\_PathSpring**フォルダを選択してください。トラッキングエラーやresiduals, reserve forceは装具追加によって増えましたか？

## VI. 動的装具を用いた歩行シミュレーション

二関節アクチュエーターの効果をシミュレーションするためのモデルが準備されています。アクチュエーターの活動量が調整できます。アクチュエーターの走行は前のセクションで用いた静的スプリングと同様にします。Geometry pathを使うと調整が可能です。

### A. 動的装具シミュレーションの実行

アクチュエーターを使用したシミュレーションは以下のステップで行います。

1. **subject01\_metabolics\_path\_actuator.osim**モデルを読み込みます。
2. **CMC**ツールから**Load...**にてセッティングファイル**walk\_Setup\_CMC\_PathActuator.xml** (CMCフォルダ) を選択してください。
3. **Navigator**パネルでモデルのpath actuator, **PlantarFlexAssist**を見つけてください。optimal forceの設定はどのようになっていますか？ optimal\_forceプロパティーでは制御シグナルの入力を調整できます。(制御シグナルが1でoptimal forceが10の時、アクチュエーターの腱は10Nを発生します。)
4. 先ほどのセクションで用いたCMCのセットアップにActuator constraintsファイルcontrols\_43p\_r.xmlが追加されています。このファイルはアクチュエーターの制御シグナル情報が保存されています。制御情報を確認するには、鉛筆アイコンを選択し、plantar flex assistのチェックをオンにしてOKを押してください。Excitation Editorが開き、アクチュエーターの活動情報が確認、修正できるようになります。(Helpを押すと説明を見ることができます)
5. 出力フォルダを**ResultsCMC\_PathActuator**に変更してください。(Output directoryを変更)
6. **Run**を押した後、Closeを押して**CMC**ツールを閉じましょう。
7. CMCが終了すれば、前記のスクリプトを使って結果を表示させてください。

## B. 制御シグナルの作成

アクチュエーターの制御シグナルを自由に設定できます。Excitation editorを使って制御シグナルを変えてみましょう。またはエクセルを使って制御シグナルを変更し、以下の手順でオープンシムに情報をインポートすることも可能です。

1. Widowsブラウザで**gait10dof18musc/PlantarFlexorAssistCurves**フォルダを探してください。フォルダ内にはPlantarFlexAssistアクチュエーターの制御を保存したstoファイル（開始時間の異なる複数の制御ファイル）があります。
2. エクセルでファイルの1つを開いてください。（**controls\_13p\_r.sto**など）
  - ファイルには2列でデータが並んでいます。1つは時間で、もう1つは活動量PlantarFlexAssist.excitation(\*ダウンロードしたファイルではPlantarFlexAssistとしか表示されませんでした)です。好きな活動パターンに変更することが可能です。
  - 時間は左の列にあり、シミュレーションでの時間経過が記されています。（ここではヒールストライクが0.6秒と1.9秒に生じる設定になっています）
  - アクチュエーター名に"\_min" or "\_max"を追加し、新しい列を加えると興奮度のシグナル情報で最大値と最小値を設定できます(例: PlantarFlexAssist.excitation\_min)。最大値や最小値を入力しなければ、次の操作で使うスクリプトは記載したシグナル値がそのまま入力されます。
3. オープンシムに戻って**Scripts>Run...**を選択し、**createControlsFromStorage.py**を実行します（OpenSim3.3/Scripts/GUI\_Scripting）。初めはステップ2で使ったファイル(**PlantarFlexorAssistCurves/controls\_13p\_r.sto**など)を選択して、**Open**を押してください。次に、CMCフォルダでファイル名を入力し（**CMC/controls\_13p\_r.xml**など）、**Save**を押してください。
  - createControlsFromStorageはバツファーです。活動量制御の最大値や最小値（シグナルを0.1足す、または減らすなどに使用）を修正してスクリプトを変更できます。これで、CMC（次のセクションA）で使うActuator constraintsファイルができました。

## VII. 他のモデルを使ってMetabolic Probeでの解析を行う

筋骨格モデルは多くの筋を有します。Metabolic Probeをすべての筋に1つずつ追加していくと時間がかかり、入力ミスもしやすいです。このため、カレントモデルのすべての筋にMetabolic Probeを設定することができるスクリプト (gait10dof18musc\_32\Scripts\addMetabolicProbes.py)を用意しています。このプローブではシミュレーションの間、モデルの各筋のエネルギー消費を計算できます。モデルの筋をすべて計算して総代謝量も算出することができます。

筋の代謝コストを計算するには速筋と遅筋の割合が必要です。gait10dof18musc\_32\ScriptsフォルダのmetabolicsSlowTwitchRatios\_gait10dof18musc.txtとmetabolicsSlowTwitchRatios\_Gait2392.txtを参照してください。これらのファイルは各筋における遅筋線維の割合を設定しており、スクリプトで用いるためのファイルです。値が-1は筋線維の割合が明確でない筋を意味しています。デフォルトは0.5となっています。

代謝プローブをGait2392モデルに追加するには次の3つの操作が必要です。：

1. オープンシムGUIからGait2392モデルを開きましょう (Models\Gait2392\_Simbody\gait2392\_simbody.osim)。
2. **AddMetabolicProbes.py**のスクリプトを実行します。
3. 各筋の線維割合を保存したファイル(**metabolicsSlowTwitchRatios\_Gait2392.txt**)を選択します。

"\_probed"がついた新しいモデルが保存・読み込みされます。同じようにgait10dof9muscモデルでも行ってみましょう。異なる筋を持つモデルでは、線維割合が書かれたテキストファイルを作るとスクリプトを使用できます。さらに詳しく知りたい場合は [doxygen](#) のページを参照してください。

## 参考文献

- Umberger, B.R., Gerritsen, K.G.M., Martin, P.E. (2003) A model of human muscle energy expenditure. *Computer Methods in Biomechanics and Biomedical Engineering*, 6(2):99–111.
- Umberger, B.R. (2010) Stance and swing phase costs in human walking. *Journal of the Royal Society Interface*, 7(50):1329–1340.
- Uchida, T.K., Hicks, J.L., Dembia, C.L., Delp, S.L. (2016) Stretching your energetic budget: how tendon compliance affects the metabolic cost of running. *PLOS ONE*, 11(3):e0150378.