

# リムレスホイール受動歩行シミュレーション サンプルプログラムの使用について

北陸先端科学技術大学院大学  
先端科学技術研究科 情報科学系  
浅野文彦研究室

## 1 はじめに

本プログラムは、下り斜面を転がるリムレスホイール（以下 RW, Fig.1 参照）の受動歩行についてのシミュレーションプログラムである。本プログラムでは MATLAB を用いて数値シミュレーションを行う。RW のモデリング、プログラムの実行方法および各プログラムファイルの詳細を以下に記す。

## 2 モデリング

### 2.1 数学モデル

本プログラムで扱う RW の仕様は以下の通りである。

- ・ 脚フレーム間の角度は  $\alpha = \pi/4$  [rad] で、8 本の脚を有する対称形状である。
- ・ 本体の質量は  $m$  [kg] とし、慣性モーメントは  $I_0$  [kg · m<sup>2</sup>] とする。
- ・ 支持脚接地点の座標を  $(x, z)$  とする。
- ・ 支持脚接地点に滑りは生じないものとする。
- ・ 時計回りを正回転とする。

### 2.2 運動方程式

一般化座標ベクトルを  $\mathbf{q} = \begin{bmatrix} x & z & \theta \end{bmatrix}^T$  とすると、運動方程式および速度拘束条件式は以下の式で示される。

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \mathbf{J}^T \boldsymbol{\lambda} \quad (1)$$

$$\mathbf{J}\dot{\mathbf{q}} = \mathbf{0}_{2 \times 1} \quad (2)$$

ここで、 $\mathbf{M}$  は慣性行列であり、 $\mathbf{h}$  は中心力、コリオリ力および重力をまとめたベクトルである。また、 $\mathbf{J}^T \boldsymbol{\lambda}$  は RW に働くホロノミック拘束力ベクトルである。各行列およびベクトルの詳細は以下の通りである。

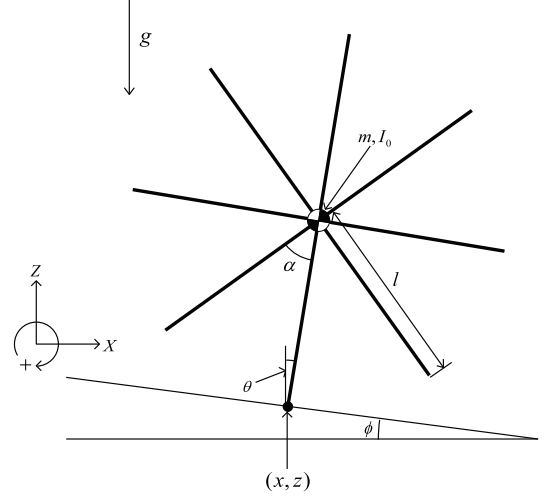


Fig.1 Model of rimless wheel

$$\mathbf{M} = \begin{bmatrix} m & 0 & ml \cos \theta \\ 0 & m & -ml \sin \theta \\ ml \cos \theta & -ml \sin \theta & ml^2 + I_0 \end{bmatrix} \quad (3)$$

$$\mathbf{h} = \begin{bmatrix} -ml\dot{\theta}^2 \sin \theta \\ mg - ml\dot{\theta}^2 \cos \theta \\ -mgl \sin \theta \end{bmatrix} \quad (4)$$

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (5)$$

### 2.3 衝突方程式

遊脚衝突直前の状態量を  $\mathbf{q}^-$ 、衝突直後の状態量を  $\mathbf{q}^+$  とする。遊脚と床面との衝突は瞬間的な完全非弾性衝突であり、衝突時に支持脚接地点に滑りが生じないと仮定すれば、衝突方程式および速度拘束条件式は

$$\mathbf{M}\dot{\mathbf{q}}^+ = \mathbf{M}\dot{\mathbf{q}}^- + \mathbf{J}_I^T \boldsymbol{\lambda} \quad (6)$$

$$\mathbf{J}_I \dot{\mathbf{q}}^+ = \mathbf{0}_{2 \times 1} \quad (7)$$

となる。また、幾何学的拘束条件により、衝突時に支持脚角度は次の式で更新される。

$$\theta^+ = \theta^- - \alpha \quad (8)$$

ただし,

$$\mathbf{J}_I = \begin{bmatrix} 1 & 0 & l \cos \theta^- - l \cos (\alpha - \theta^-) \\ 0 & 1 & -l \sin \theta^- - l \sin (\alpha - \theta^-) \end{bmatrix} \quad (9)$$

である.

また, 式 (6) および (7) について解くことにより, 衝突直後の速度ベクトル  $\dot{\mathbf{q}}^+$  は

$$\dot{\mathbf{q}}^+ = (\mathbf{I}_3 - \mathbf{M}^{-1} \mathbf{J}_I^T \mathbf{X}^{-1} \mathbf{J}_I) \dot{\mathbf{q}}^- \quad (10)$$

$$\mathbf{X} := \mathbf{J}_I \mathbf{M}^{-1} \mathbf{J}_I^T \quad (11)$$

となる.

### 3 実行方法および各ファイルの詳細

#### 3.1 プログラムの実行

本プログラムは MATLAB 上でシミュレーションを実行する. メインプログラムファイル `<main.m>` を MATLAB エディタ上で開き, 上部ツールバーの実行ボタンを押すことでシミュレーションを開始する.

#### 3.2 各ファイルの詳細

同梱されている各ファイルに関して, 詳細を以下に示す.

##### `<main.m>`

本プログラムのメインファイル. シミュレーションに係る各種パラメータの設定をはじめ, ode45 ソルバーによる運動方程式の数値積分やシミュレーション結果の出力を行う.

##### `<rimless.m>`

RW の運動方程式に基づき, 状態量の時間変化率を返す関数. これを ode ソルバーによって数値積分することで, 状態量の時間変化について得ることができる.

##### `<detectCollision.m>`

遊脚の着地を検出する関数. この関数によって検出された遊脚の着地をイベントとして, ode ソルバーの数値積分を打ち切る.

##### `<changeLeg.m>`

脚交換を行う関数. 衝突方程式に基づいて状態量を更新し, 支持脚の交換を行う.

##### `<createFigure.m>`

数値シミュレーションの結果をグラフへプロットする関数. Figure ウィンドウとして各種プロットを表示することに加え, 同プロットを fig ディレクトリ内へ保存する.

##### `<createAnimation.m>`

数値シミュレーションの結果をアニメーション表示する関数. RW の挙動が確認できる簡易的なアニメーションを表示することに加え, 同アニメーションを movie ディレクトリ内へ保存する.

## 4 おわりに

### 4.1 免責事項

本マニュアルおよび同梱のプログラムは注意を払った上で作成を行っておりますが, 情報の信頼性等を保証するものではありません. また, 当該プログラムの利用により生じた損害や損失において, 当方はいかなる責任も負いかねます.

### 4.2 著作権

本マニュアルおよび同梱のプログラムの著作権は北陸先端科学技術大学院大学 先端科学技術研究科 情報科学系 浅野文彦研究室に属します. 本文面やプログラム等の無断転載, ならびに営利目的での利用はご遠慮ください.

© 2019 F. Asano Laboratory, All Rights Reserved.

### 4.3 作成者

本マニュアルおよび同梱のプログラムに誤りを見つけた際や, ご不明な点がございましたら下記メールアドレスまでご連絡ください.

小林 聖弥 (Seiya KOBAYASHI)

Mail: seiya-kobayashi@jaist.ac.jp

### 4.4 改訂履歴

2012 年 02 月 23 日 第 1 版 第 1 刷発行

2012 年 10 月 01 日 第 1 版 第 2 刷発行

2019 年 03 月 15 日 第 2 版 第 1 刷発行