Multibody simulation

bMSd toolbox

Dimitar Dimitrov

Örebro University

September 25, 2011

Main points covered

- how to use the bMSd toolbox

Installation

What does it do?

`bMSd` (Basic Multibody Simulator - Derived) is a Matlab toolbox that can be used to perform kinematic and dynamic simulation and analysis of open-loop manipulator systems with a free-floating base.

Is it fast?

The simulator is very inefficiently implemented. It is mainly used for educational purposes.

How to install?

Download it from the course website, navigate to its directory ($e.g.$, `bMSd`) and type `setup_bMSd`. This will include in the Matlab's path a number of directories that contain files that you could use.

Notation

parent-child structure

- link i will be denoted by L_i
- the **Parent Link** (PL) of L_i is the link immediately below it in the kinematic tree (the one that directly "supports" L_i)
- every link has only one PL (the PL for L_1 is the "environment")
- a link can be a PL of zero, one or many links
- the PL of L_i will be denoted by $L_{\mathcal{P}(i)}$

- **Input Join** (IJ) for L_i is the joint connecting L_i and $L_{\mathcal{P}(i)}$
- only two types of joints can be defined: **P**rismatic and **R**evolute. **P** and **R** joints can be used to model many types of joints
- the base (L_1) could be either rigidly connected to the "environment" (0 DoF) or can be free-floating (6 DoF)

- the "environment" has number 0
- the base link has number 1 (L_1), the next link has number 2 etc.
- J_i is an IJ for L_{i+1}

Coordinate frames

see figure on next slide

- $\{\mathcal{L}_0\}$ denotes the **world frame**
- $\{\mathcal{L}_1\}$ denotes the frame whose origin coincides with the CoM of L_1
- $\{\mathcal{L}_i\}$ $(i = 2, \dots, n+1)$ is a frame associated with L_i, whose origin coincides with the input joint of L_i $(i.e.$, joint $J_{i-1})$
- the z axis of $\{\mathcal{L}_i\}$ $(i = 2, \dots, n+1)$ is assumed to be the axis of rotation/translation of J_{i-1}

The relative orientation between frames $\{\mathcal{L}_i\}$ and $\{\mathcal{L}_k\}$ will be represented by the 3×3 rotation matrix ${}^i_k\boldsymbol{R}$. Hence, if ${}^k\boldsymbol{v}$ is a vector expressed in $\{\mathcal{L}_k\}$ (to be denoted by ${}^k\boldsymbol{v} \in \{\mathcal{L}_k\}$), then ${}^i_k\boldsymbol{R}^k\boldsymbol{v} \in \{\mathcal{L}_i\}$.

when the left superscript is 0, we will use $\boldsymbol{R}_3$ instead of ${}^0_3\boldsymbol{R}$

- $\boldsymbol{R}_3$ is the rotation matrix that sends vectors from frame $\{\mathcal{L}_3\}$ to the world frame
- $\boldsymbol{R}_3^T$ is the rotation matrix that sends vectors from the world frame to $\{\mathcal{L}_3\}$

Figure (coordinate frames)

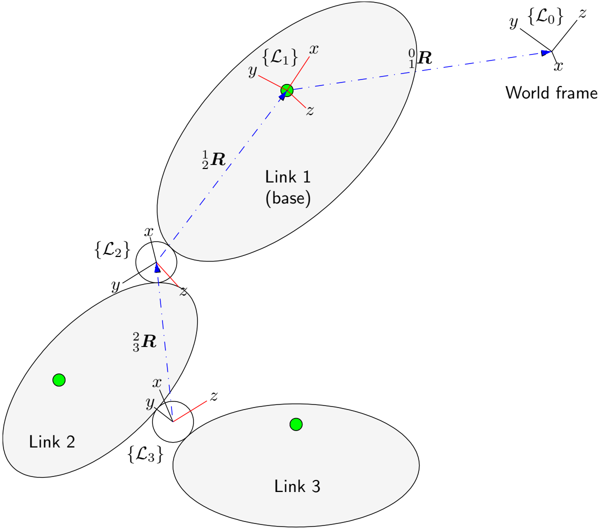

Figure: Placement of coordinate frames. White circles represent joints, gray ellipses represent links. Link center of mass is denoted by a green dot.

the relative position of the body-fixed frames $\{\mathcal{L}_i\}$ is defined as follows

see figure on next slide

- $\boldsymbol{j}_i^t$ is the vector from the CoM of $L_{\mathcal{P}(i+1)}$ **to** J_i expressed in $\{\mathcal{L}_{\mathcal{P}(i+1)}\}$
- $\boldsymbol{j}_i^f$ vector **from** J_i to CoM of L_{i+1} expressed in $\{\mathcal{L}_{i+1}\}$

For the system in the figure on the next slide

$$\left(\boldsymbol{j}_4^t + \tfrac{1}{5}\boldsymbol{R}\boldsymbol{j}_4^f\right) \in \{\mathcal{L}_1\}.$$

Note that, it would not make sense to write $\boldsymbol{j}_4^t + \boldsymbol{j}_4^f$, since the two vectors are expressed in different frames, $i.e.$,

- $\boldsymbol{j}_4^t \in \{\mathcal{L}_1\}$
- $\boldsymbol{j}_4^f \in \{\mathcal{L}_4\}$

As another example, $\boldsymbol{j}_2^f$ is the vector from J_2 to the CoM of L_3, expressed in frame $\{\mathcal{L}_3\}$.

Figure (connectivity)

Connectivity

$$\mathcal{P} = \left[\begin{array}{ccccc} 0 & 1 & 2 & 1 & 1 \end{array}\right]$$

Hence,

- L_1 is connected to the environment ($L_{\mathcal{P}(1)} = 0$)
- L_2 is connected to L_1 ($L_{\mathcal{P}(2)} = 1$)
- L_3 is connected to L_2 ($L_{\mathcal{P}(3)} = 2$)
- L_4 is connected to L_1 ($L_{\mathcal{P}(4)} = 1$)
- L_5 is connected to L_1 ($L_{\mathcal{P}(5)} = 1$)

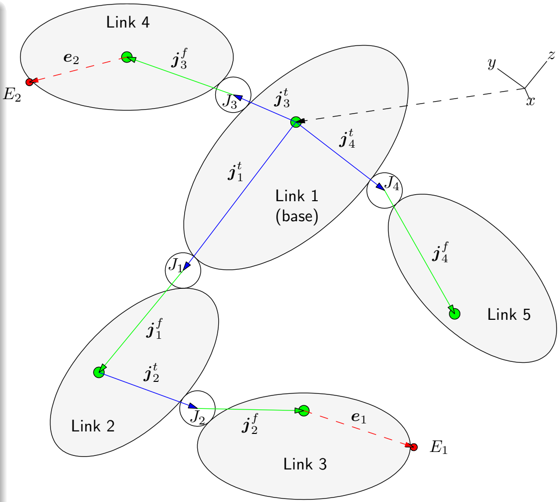

Figure: Multibody system connectivity and dimensions. White circles represent joints, gray ellipses represent links. Link center of mass is denoted by a green dot. The red dots stand for end-effectors.

Creating a model in bMSd

geometric parameters

define a Matlab *structure* (let us call it `SP` for System Parameters) containing the following fields

$\mathcal{P}$	`SP.C`	connectivity vector
$\boldsymbol{j}_i^t \in \{L_{\mathcal{P}(i+1)}\}$	`SP.J(i).t`	vector to J_i from CoM of $L_{\mathcal{P}(i+1)}$
$\boldsymbol{j}_i^f \in \{L_{i+1}\}$	`SP.J(i).f`	vector from J_i to CoM of L_{i+1}
J_i^{type}	`SP.J(i).type`	type of joint i `SP.J(i).type = 'P'` - Prismatic `SP.J(i).type = 'R'` - Revolute
$\mathcal{E}_i$	`SP.J(i).rpy`	Euler angles $x \to y \to z$ (current axis)

$\mathcal{E}_i = \{\alpha_i, \beta_i, \gamma_i\}$ define the orientation of $\{\mathcal{L}_i\}$ w.r.t. $\{\mathcal{L}_{\mathcal{P}(i)}\}$. The rotation matrix corresponding to $\mathcal{E}_i$ is given by $(j = \mathcal{P}(i))$

$$ {}^j_i\boldsymbol{R} = \boldsymbol{R}_x(\alpha_i)\boldsymbol{R}_y(\beta_i)\boldsymbol{R}_z(\gamma_i). $$

Hence, for a vector $\boldsymbol{v} \in \{\mathcal{L}_i\}$, ${}^j_i\boldsymbol{R}\boldsymbol{v} \in \{\mathcal{L}_{\mathcal{P}(i)}\}$.
See files `/rotation/rpy2R.m`, `./examples/test_rot_seq.m`

Creating a model in bMSd

dynamic parameters

m_i	`SP.L(i).m`	mass of L_i
$\mathcal{I}_i$	`SP.L(i).I`	inertia tensor of L_i

$\mathcal{I}_i \in \mathbb{R}^{3\times 3}$ is the inertia matrix of L_i about its CoM, expressed in $\{\mathcal{L}_i\}$.

recall that

the point about which $\mathcal{I}_i$ is computed, and the frame where it is expressed are two different things

Creating a model in bMSd

other parameters

`SP.mode`	*mode* of the base (fixed of free-floating)
	`SP.mode = 0` - base is free-floating (6DoF)
	`SP.mode = 1` - base is fixed to the environment (0DoF)
`SP.bN(i)`	is the PL of end-effector i
`SP.bP(:,i)`	defines the position of the i^{th} end-effector

`SP.bP(:,i)` is a vector from the CoM of link `SP.bN(i)` to the end-effector, expressed in the local frame in link `SP.bN(i)`

System variables

The system variables are defined in a Matlab *structure* (denoted by `SV` for System Variables) containing the following fields

`SV.q(i)`	angle of joint i
`SV.dq(i)`	velocity of joint i
`SV.ddq(i)`	acceleration of joint i
`SV.tau(i)`	torque of joint i (applied by motor)
`SV.L(i).R`	rotation matrix of $\{\mathcal{L}_i\}$ w.r.t. $\{\mathcal{L}_0\}$
`SV.L(i).Q`	quaternion of $\{\mathcal{L}_i\}$ w.r.t. $\{\mathcal{L}_0\}$
`SV.L(i).p`	position of CoM of L_i in $\{\mathcal{L}_0\}$
`SV.L(i).v`	linear velocity of CoM of L_i in $\{\mathcal{L}_0\}$
`SV.L(i).dv`	linear acceleration of CoM of L_i in $\{\mathcal{L}_0\}$
`SV.L(i).w`	angular velocity of L_i in $\{\mathcal{L}_0\}$
`SV.L(i).dw`	angular acceleration of L_i in $\{\mathcal{L}_0\}$
`SV.L(i).T`	torque applied to L_i in $\{\mathcal{L}_0\}$
`SV.L(i).F`	force applied to the CoM of L_i in $\{\mathcal{L}_0\}$

User input

n is the number of joints

for $i = 1, 2, \ldots, n$

- `SV.q(i)`
- `SV.dq(i)`
- `SV.tau(i)`

for $i = 1, 2, \ldots, n+1$

- `SV.L(i).F`
- `SV.L(i).T`

the state of the base can be set using

- `SV.L(1).R`
- `SV.L(1).p`
- `SV.L(1).v`
- `SV.L(1).w`

Example (planar system)

definition of system structure

```
SP.C = [0 1 2]; % connectivity
SP.n = length(SP.C)-1; % 2 joints
SP.mode = 1; % fixed-base system
```

definition of joints

```
% Joint 1
SP.J(1).t    = [ 0.5  0.0  0.0 ]';
SP.J(1).f    = [ 0.5  0.0  0.0 ]';
SP.J(1).rpy  = [ 0.0  0.0  0.0 ]';
SP.J(1).type = 'R';

% Joint 2
SP.J(2).t    = [ 0.5  0.0  0.0 ]';
SP.J(2).f    = [ 0.5  0.0  0.0 ]';
SP.J(2).rpy  = [ 0.0  0.0  0.0 ]';
SP.J(2).type = 'R';
```

definition of links

```
% Link 1
SP.L(1).m = 1;      % mass
SP.L(1).I = eye(3); % inertia matrix

% Link 2
SP.L(2).m = 1;
SP.L(2).I = eye(3);

% Link 3
SP.L(3).m = 1;
SP.L(3).I = eye(3);
```

definition of end-effector

```
SP.bN = 3; % only one end-effector with parent link 3
SP.bP = [0.5 0.0 0.0]';
```