

Robotics Toolbox Matlab

1. Create a rigid body object.

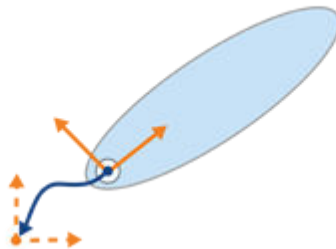
```
body1 = rigidBody('body1');
```



2. Create a joint and assign it to the rigid body. Define the home position property of the joint, HomePosition. Set the joint-to-parent transform using a homogeneous transformation, tform. Use the `trvec2tform` function to convert from a translation vector to a homogenous transformation. ChildToJointTransform is set to an identity matrix.

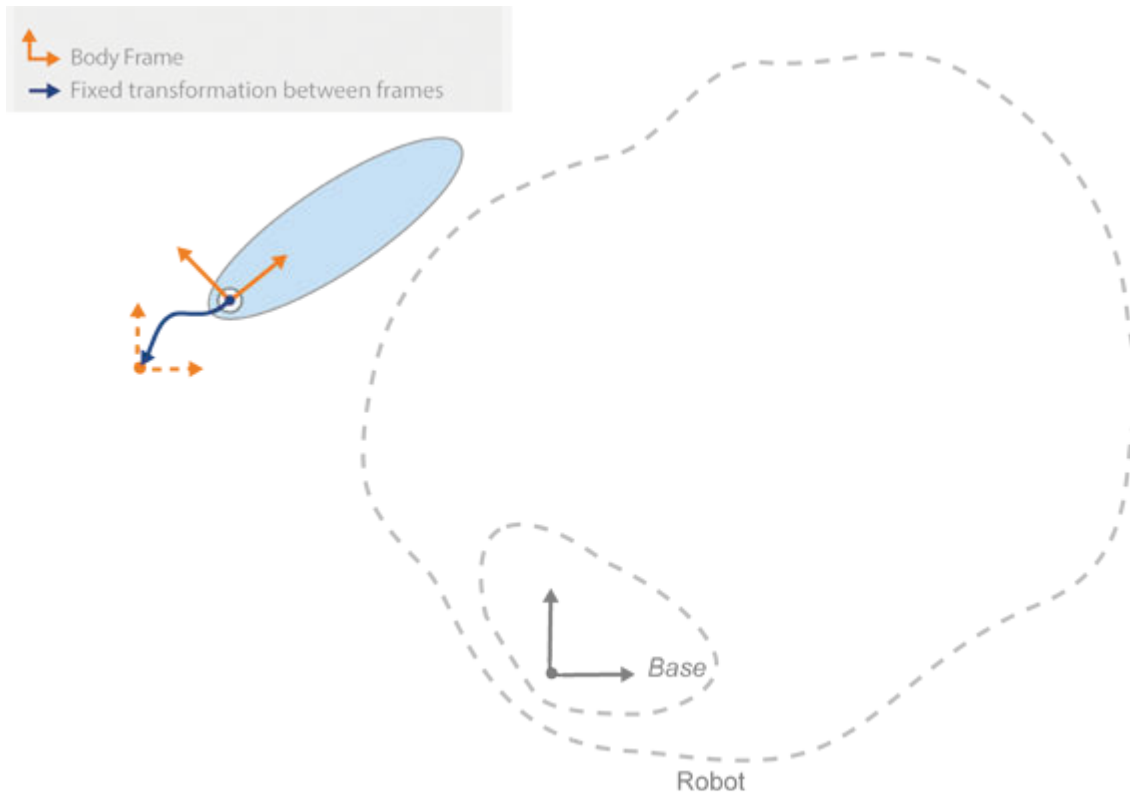
```
jnt1 = rigidBodyJoint('jnt1','revolute');  
jnt1.HomePosition = pi/4;  
tform = trvec2tform([0.25, 0.25, 0]); % User defined  
setFixedTransform(jnt1,tform);  
body1.Joint = jnt1;
```

 Body Frame
 Fixed transformation between frames



3. Create a rigid body tree.

```
robot = rigidBodyTree;
```



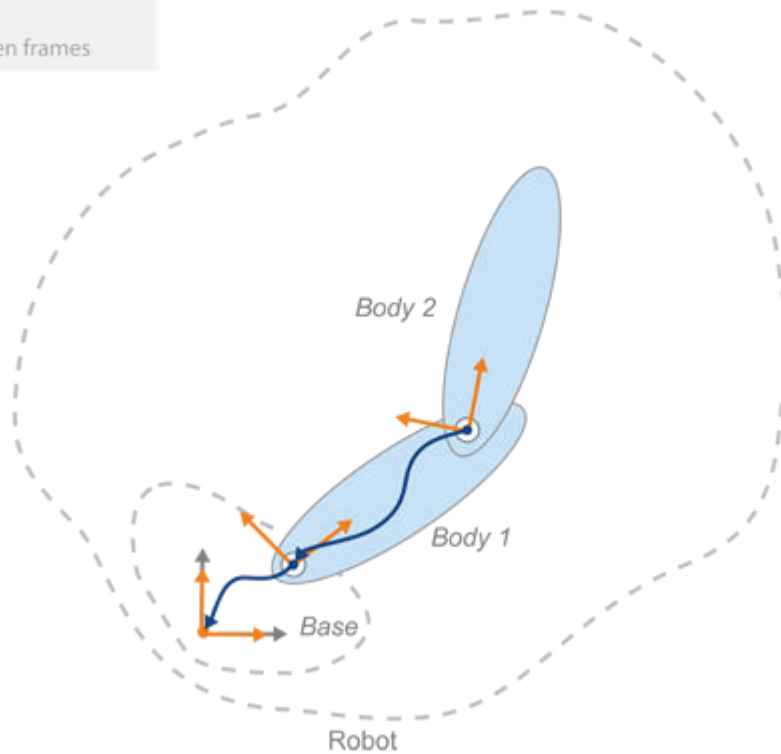
4. Add the first body to the tree. Specify that you are attaching it to the base of the tree. The fixed transform defined previously is from the base (parent) to the first body.

```
addBody(robot, body1, 'base')
```

5. Create a second body. Define properties of this body and attach it to the first rigid body. Define the transformation relative to the previous body frame.

```
body2 = rigidBody('body2');
jnt2 = rigidBodyJoint('jnt2', 'revolute');
jnt2.HomePosition = pi/6; % User defined
tform2 = trvec2tform([1, 0, 0]); % User defined
setFixedTransform(jnt2, tform2);
body2.Joint = jnt2;
addBody(robot, body2, 'body1'); % Add body2 to body1
```

 Body Frame
 Fixed transformation between frames



6. Add other bodies. Attach body 3 and 4 to body 2.

```
body3 = rigidBody('body3');
body4 = rigidBody('body4');
jnt3 = rigidBodyJoint('jnt3','revolute');
jnt4 = rigidBodyJoint('jnt4','revolute');
tform3 = trvec2tform([0.6, -0.1, 0])*eul2tform([-pi/2, 0, 0]); % User defined
tform4 = trvec2tform([1, 0, 0]); % User defined
setFixedTransform(jnt3,tform3);
setFixedTransform(jnt4,tform4);
jnt3.HomePosition = pi/4; % User defined
body3.Joint = jnt3
```

```
body3 =
  rigidBody with properties:

    Name: 'body3'
    Joint: [1x1 rigidBodyJoint]
    Mass: 1
    CenterOfMass: [0 0 0]
    Inertia: [1 1 1 0 0 0]
    Children: {1x0 cell}
    Visuals: {}
    Collisions: {}
    FrameNames: {'body3'}
    ParentFrame: ''
```

```
body4.Joint = jnt4
```

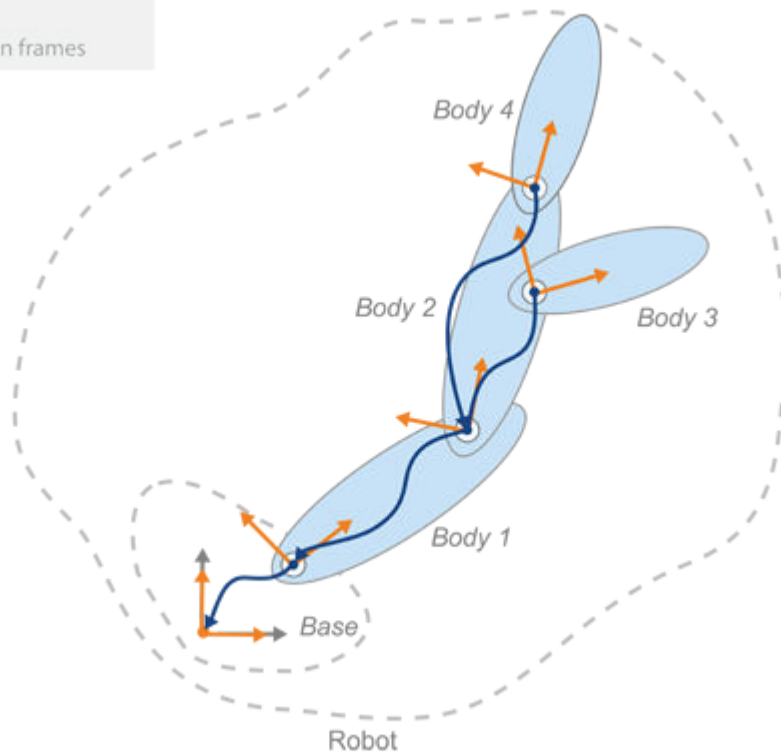
```
body4 =
```

rigidBody with properties:

```
Name: 'body4'  
Joint: [1x1 rigidBodyJoint]  
Mass: 1  
CenterOfMass: [0 0 0]  
Inertia: [1 1 1 0 0 0]  
Children: {1x0 cell}  
Visuals: {}  
Collisions: {}  
FrameNames: {'body4'}  
ParentFrame: ''
```

```
addBody(robot, body3, 'body2'); % Add body3 to body2  
addBody(robot, body4, 'body2'); % Add body4 to body2
```

Body Frame
Fixed transformation between frames



7. If you have a specific end effector that you care about for control, define it as a rigid body with a fixed joint. For this robot, add an end effector to body4 so that you can get transformations for it.

```
bodyEndEffector = rigidBody('endeffector');  
tform5 = trvec2tform([0.5, 0, 0]); % User defined  
setFixedTransform(bodyEndEffector.Joint, tform5);  
addBody(robot, bodyEndEffector, 'body4');
```

8. Now that you have created your robot, you can generate robot configurations. With a given configuration, you can also get a transformation between two body frames using `getTransform`. Get a transformation from the end effector to the base.

```
% config = randomConfiguration(robot)
config = homeConfiguration(robot)
```

```
config = 1×4 struct
```

Fields	JointName	JointPosition
1	'jnt1'	0.7854
2	'jnt2'	0.5236
3	'jnt3'	0.7854
4	'jnt4'	0

```
% % Assign custom values (in radians for revolute, meters for prismatic
joints)
```

```
config(1).JointPosition = 0.5; % Joint 1
config(2).JointPosition = -0.3; % Joint 2
config(3).JointPosition = 1.2; % Joint 3
config(4).JointPosition = 0.0; % Joint 4
```

```
tform = getTransform(robot,config,'endeffector','base')
```

```
tform = 4×4
```

```
1.0000    0    0    2.7500
0    1.0000    0    0.2500
0    0    1.0000    0
0    0    0    1.0000
```

9. Finally, you can use `showdetails` to look at the robot you built. Verify that the joint types are correct.

```
showdetails(robot)
```

```
-----
Robot: (5 bodies)
```

Idx	Body Name	Joint Name	Joint Type	Parent Name(Idx)	Children
---	-----	-----	-----	-----	-----
1	body1	jnt1	revolute	base(0)	body2(2)
2	body2	jnt2	revolute	body1(1)	body3(3)
3	body3	jnt3	revolute	body2(2)	
4	body4	jnt4	revolute	body2(2)	endeffec
5	endeffector	endeffector_jnt	fixed	body4(4)	

```
-----
```