Lecture 2 - Simulation of Rigid Body Dynamics

Julius Martensen

DFKI Bremen & Universität Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de

Rigid Body Dynamics

Robots as kinematic chains

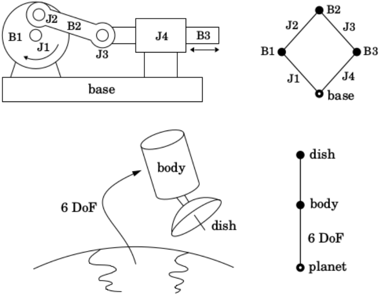


Figure: Taken from Featherstone, *Rigid Body Dynamics Algorithms*

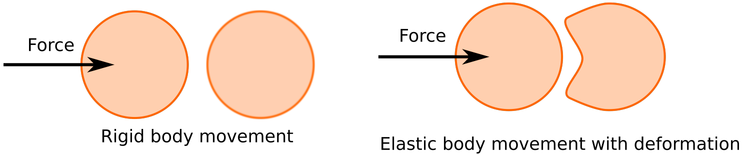
Equations of Motion

In general, the motion of a mechanical structure can be described as a set of nonlinear differential equations

$$M_q \ddot{q} + C(q, \dot{q}) = \sum \tau$$

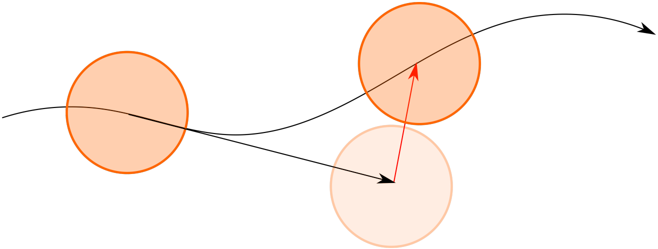
Where $M_q \in \mathbb{R}^{m \times m}$ is the generalized inertia matrix, $C \in \mathbb{R}^m$ is the vector of centrifugal and gravitational forces and $\tau \in \mathbb{R}^m$ are the external forces.

Rigid Body Movement



Constraints

A constrained particle



Description of Constraints

The trajectory is constrained via a force which arise due to algebraic constraints:

$$\begin{aligned} \textit{Explicit}: &\; x = f(q) \\ \textit{Implicit}: &\; g(q) = 0 \end{aligned}$$

Which are called holonomic constraints, iff they are only depending on the configuration.

Implementation of Constraints

A system under explicit constraints can then be described as:

$$\begin{aligned} M_q\ddot{q} + C(q,\dot{q}) &= \tau + \tau_C \\ \tau_C &= \frac{df(q)}{dq}^T \lambda \\ &= J^T \lambda \end{aligned}$$

Which is sufficient to the principle of virtual power.

Implementation of Constraints

Which leads to the following system of equations

$$\begin{bmatrix} M_q & J^T \\ J & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \tau - C(q, \dot{q}) \\ k \end{bmatrix}$$

The system is solveable, iff the coefficient Matrix is nonsingular.
This is not always the case!

Constraints

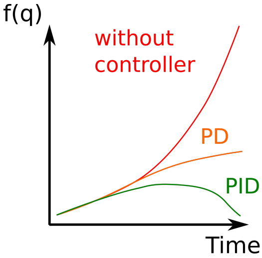
A common strategy to get the constraint force is to derive the explicit constraint w.r.t. time:

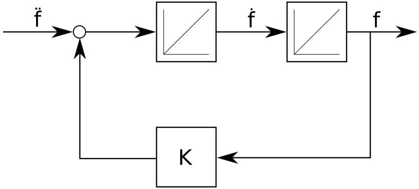
$$
\begin{aligned}
g(q) &= 0 \\
\frac{dg}{dt} &= \frac{dg}{dq}\dot{q} = 0 \\
\frac{d^2g}{dt^2} &= \frac{d}{dt}\left[\frac{dg}{dq}\dot{q}\right] \quad = \dots + \frac{dg}{dq}\ddot{q} = 0
\end{aligned}
$$

And to solve for $\ddot{q}$.

Constraints

This may lead to an increasing error, since we have a numeric precision on acceleration level! To compensate, most of the time a controller is implemented based on [BAUMGARTE].





Constraints

Caveats

- The controller parameter can often be adjusted within simulation software.
- These techniques reduces the error, but changes the forces in the system.
- A change in the parameter set can change the whole simulation result!

Caveats

- The controller parameter can often be adjusted within simulation software.
- These techniques reduces the error, but changes the forces in the system.
- A change in the parameter set can change the whole simulation result!

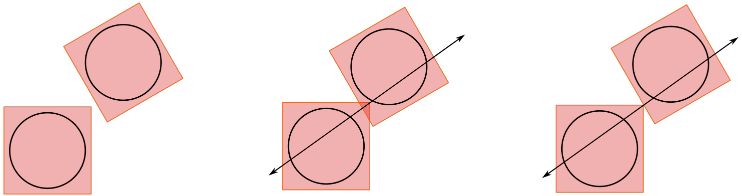
Constraints

Caveats

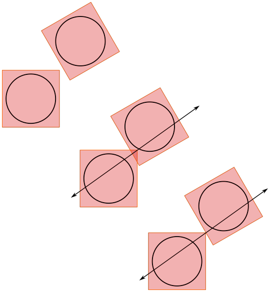
- The controller parameter can often be adjusted within simulation software.
- These techniques reduces the error, but changes the forces in the system.
- A change in the parameter set can change the whole simulation result!

Collisions and Friction

Collisions

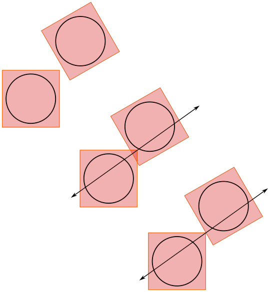


Implementation of Collisions



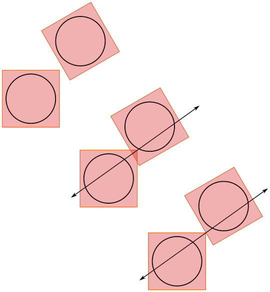
- Check for collision
- Find collision points
- Check for penetration depth
- Apply force proportional to penetration depth

Implementation of Collisions



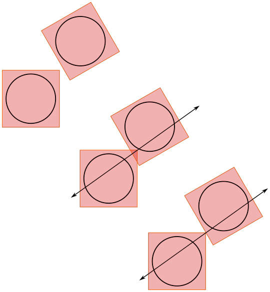
- Check for collision
- Find collision points
- Check for penetration depth
- Apply force proportional to penetration depth

Implementation of Collisions



- Check for collision
- Find collision points
- Check for penetration depth
- Apply force proportional to penetration depth

Implementation of Collisions



- Check for collision
- Find collision points
- Check for penetration depth
- Apply force proportional to penetration depth

Collisions and Friction

Caveats Collision

- Collisions depend strongly on simulation (hyper-) parameters!
- The resulting force are at best a linearization of the real forces!
- Most engines "just" calculate point to point contacts!
- If primitive shapes are used, the behavior is strongly influenced by the shape.

Caveats Collision

- Collisions depend strongly on simulation (hyper-) parameters!
- The resulting force are at best a linearization of the real forces!
- Most engines "just" calculate point to point contacts!
- If primitive shapes are used, the behavior is strongly influenced by the shape.

Collisions and Friction

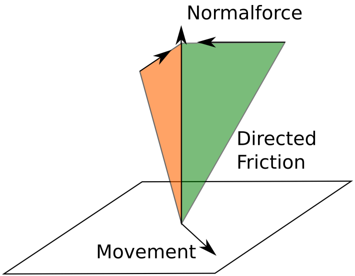
Caveats Collision

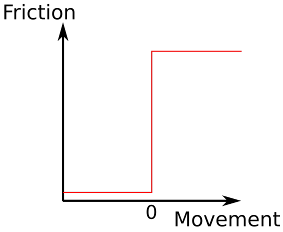
- Collisions depend strongly on simulation (hyper-) parameters!
- The resulting force are at best a linearization of the real forces!
- Most engines "just" calculate point to point contacts!
- If primitive shapes are used, the behavior is strongly influenced by the shape.

Caveats Collision

- Collisions depend strongly on simulation (hyper-) parameters!
- The resulting force are at best a linearization of the real forces!
- Most engines "just" calculate point to point contacts!
- If primitive shapes are used, the behavior is strongly influenced by the shape.

Friction


Normalforce
Directed
Friction
Movement


Friction
0
Movement

Friction Models

Friction in general is described with a simple Coloumb friction model. This model is not sufficient for all use cases!

$$
\begin{aligned}
Coloumb: \;\; & F_C = -\mu \; F_N \; sign(\dot{x}) \\
Viscous: \;\; & F_V = F_C \; sat(k\dot{x}) \\
Dankowicz: \;\; & F_D = \frac{F_{Max}}{\delta} \; z \\
& \dot{z} = \dot{x} \left[1 - \frac{z}{\delta} \; sign(\dot{x}) \right]
\end{aligned}
$$

Caveats Friction

- Friction is depending on contact force!
- Most likely, only a simple friction model is implemented.
- Sometimes, there is not directional friction.
- If primitive shapes are used, the behavior is strongly influenced by the shape.

Caveats Friction

- Friction is depending on contact force!
- Most likely, only a simple friction model is implemented.
- Sometimes, there is not directional friction.
- If primitive shapes are used, the behavior is strongly influenced by the shape.

Caveats Friction

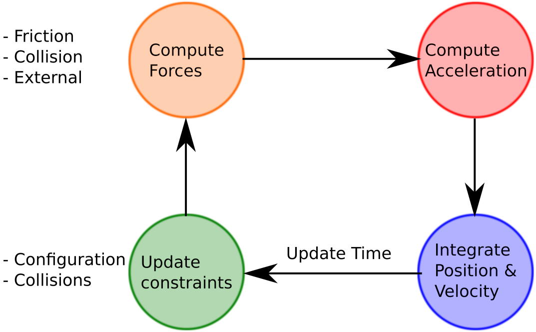
- Friction is depending on contact force!
- Most likely, only a simple friction model is implemented.
- Sometimes, there is not directional friction.
- If primitive shapes are used, the behavior is strongly influenced by the shape.

Caveats Friction

- Friction is depending on contact force!
- Most likely, only a simple friction model is implemented.
- Sometimes, there is not directional friction.
- If primitive shapes are used, the behavior is strongly influenced by the shape.

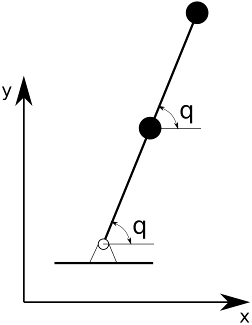
Collisions and Friction

Algorithm for Multibody Dynamics



URDF

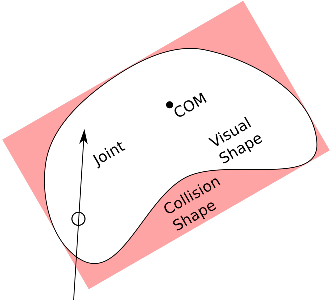
Universal Robot Description Format



```xml
<?xml version="1.0"?>
<!-- created with Phobos 0.7 -->
 <robot name="pendulum">

  <link name="revolute">
   <inertial>
    <origin xyz="-9.54545 0 0" rpy="0 0 0"/>
    <mass value="11.0"/>
    <inertia ixx="1.00167" ixy="0" ixz="0" iyy="32.06144" iyz="0" izz="32.06144"/>
   </inertial>
   <visual name="link">
    <origin xyz="-5.0 0 0" rpy="0 -1.5708 0"/>
    <geometry>
     <box size="0.1 0.1 10.0"/>
    </geometry>
   </visual>
   <visual name="mass">
    <origin xyz="-10.0 0 0" rpy="0 -1.5708 0"/>
    <geometry>
     <sphere radius="0.5"/>
    </geometry>
   </visual>
   <collision name="link.001">
    <origin xyz="-5.0 0 0" rpy="0 -1.5708 0"/>
    <geometry>
     <box size="0.1 0.1 10.0"/>
    </geometry>
   </collision>
   <collision name="mass.001">
    <origin xyz="-10.0 0 0" rpy="0 -1.5708 0"/>
    <geometry>
     <box size="1.0 1.0 1.0"/>
    </geometry>
   </collision>
  </link>

  <link name="root">
  </link>

  <joint name="revolute" type="revolute">
   <origin xyz="0 0 0" rpy="0 1.5708 0"/>
   <parent link="root"/>
   <child link="revolute"/>
   <limit effort="1000.0" velocity="17.45329"/>
  </joint>

 </robot>
```

Links

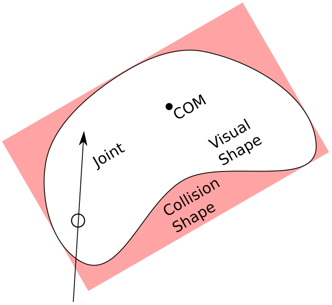
- Name
- Inertial information
  Mass, Inertia, Pose
- Visual information
  Origin, Geometry, Pose
- Collision information
  Origin. Geometry, Pose



URDF

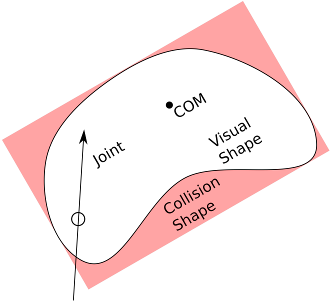
Links

- Name
- Inertial information
  Mass, Inertia, Pose
- Visual information
  Origin, Geometry, Pose
- Collision information
  Origin, Geometry, Pose



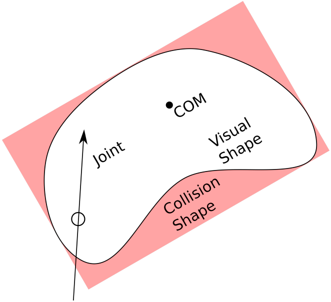
Links

- Name
- Inertial information
  Mass, Inertia, Pose
- Visual information
  Origin, Geometry, Pose
- Collision information
  Origin. Geometry, Pose



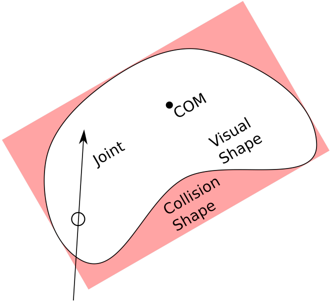
Links

- Name
- Inertial information
  Mass, Inertia, Pose
- Visual information
  Origin, Geometry, Pose
- Collision information
  Origin. Geometry, Pose



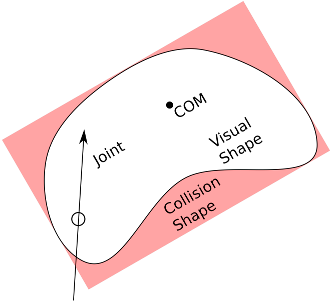
Joints

- Name
- Pose
  relative to Parent
- Parent Link
- Child Link
- Limits
  Position, Velocity, Effort



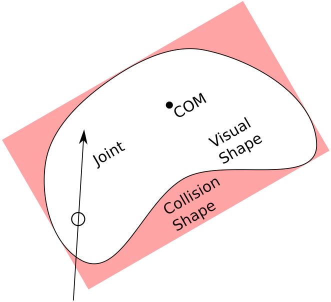
Joints

- Name
- Pose
  relative to Parent
- Parent Link
- Child Link
- Limits
  Position, Velocity, Effort



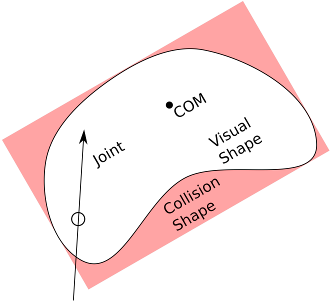
Joints

- Name
- Pose
  relative to Parent
- Parent Link
- Child Link
- Limits
  Position, Velocity, Effort



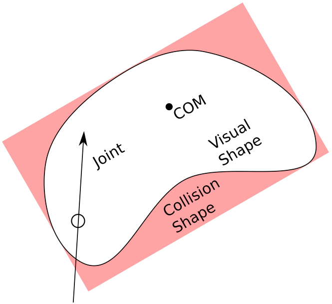
Joints

- Name
- Pose
  relative to Parent
- Parent Link
- Child Link
- Limits
  Position, Velocity, Effort



Joints

- Name
- Pose
  relative to Parent
- Parent Link
- Child Link
- Limits
  Position, Velocity, Effort



Phobos - A WYSIWYG Robot Modeling Tool

Bibliography

References I

Adamy, Jrgen. *Nichtlineare Regelungen*. de. OCLC: 434518094. Berlin: Springer, 2009. ISBN: 978-3-642-00793-4 978-3-642-00794-1.

Featherstone, Roy. *Rigid Body Dynamics Algorithms*. en. Springer US, 2008. ISBN: 978-0-387-74314-1. URL: //www.springer.com/de/book/9780387743141 (visited on 11/14/2018).

Hairer, Ernst, Syvert P. Nrsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. en. 2nd ed. Springer Series in Computational Mathematics, Springer Ser.Comp.Mathem. Hairer,E.:Solving Ordinary Diff. Berlin Heidelberg: Springer-Verlag, 1993. ISBN: 978-3-540-56670-0. URL: //www.springer.com/de/book/9783540566700 (visited

References II

Hairer, Ernst, Michel Roche, and Christian Lubich. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. en. Vol. 1409. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989. ISBN: 978-3-540-51860-0 978-3-540-46832-5. DOI: 10.1007/BFb0093947. URL: http://link.springer.com/10.1007/BFb0093947 (visited on 11/14/2018).

References III

- Hairer, Ernst and Gerhard Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. en. 2nd ed. Springer Series in Computational Mathematics, Springer Ser.Comp.Mathem. Hairer,E.:Solving Ordinary Diff. Berlin Heidelberg: Springer-Verlag, 1996. ISBN: 978-3-540-60452-5. URL: //www.springer.com/de/book/9783540604525 (visited on 11/14/2018).
- Khalil, Hassan K. *Nonlinear control*. Boston: Pearson, 2015. ISBN: 978-0-13-349926-1. URL: http://katalog.suub.uni-bremen.de/DB=1/LNG=DU/CMD?ACT=SRCHA&IKT=8000&TRM=1466249811*.
- Lynch, Kevin M. and Frank C. Park. *Modern Robotics*. en. Google-Books-ID: 5NzFDgAAQBAJ. Cambridge University

References IV

Strogatz, Steven H. *Nonlinear Dynamics and Chaos : With Applications to Physics, Biology, Chemistry, and Engineering*. en. CRC Press, May 2018. ISBN: 978-0-429-96111-3. DOI: 10.1201/9780429492563. URL: https://www.taylorfrancis.com/books/9780429961113 (visited on 11/14/2018).