目录

1. What is Multibody Dynamics?

2. Modeling Elements

3.Properties and Kinematics of Rigid Bodies

4. Constraint Modeling

5. Simulation of free-floating Rigid Bodies

6.Constraint forces and simulation

7.Stabilization

8.Impulse-based simulation of constrained Rigid Bodies

9.Simulation of constrained Rigid Bodies with contacts

10.Simulation of Constrained Rigid Bodies with Friction

MBD

Multibody Dynamics (MBD) presentation logic framework

1. What is Multibody Dynamics?

ref: C10.a

- MBD studies systems composed of multiple rigid bodies interconnected via joints or contacts.
- The objective is to analyze the motion of these bodies under applied forces and kinematic constraints.

MBD = Multiple Rigid Body Dynamics + Joints, constraints or contacts

- Simulation Model： describe
  - properties of each individual Rigid Body （mass, inertia, center of gravity, shape）
  - kinematic coupling between Rigid Bodies
  - additional actors/sensors
- Simulator：
  - Provides solvers/integrators

Simulation system structure and process

- Inverse Dynamics Solver
- Forward Dynamics Integrator

（We will introduce next）

2. Modeling Elements

ref: C10.b

- **Coordinate Representations**:
  - *Generalized Coordinates*: Minimal set to describe system configuration.
  - *Maximal Coordinates*: Full position and orientation of each body (preferred for contact handling).

{

why we use *Maximal Coordinates* for this thesis?

Because it is **more general, flexible, and robust** for real-world multibody systems — especially when dealing with **contacts and collisions.**

Generalized coordinates are efficient tools in theory, but in engineering practice, they lack flexibility and cannot handle complex, dynamic, and interactive multi-body systems. This is the fundamental reason why we abandon generalized coordinates and choose the maximum coordinate system.

Maximal coordinates may be redundant, but that redundancy gives us the power to simulate anything — from soft contact to space robotics — with modularity, realism, and numerical stability.

}

- Fundamental dynamic equations   based **Momentum conservation**

Newtons laws of motion for Rigid Body

$$\underline{p}_i = m_i \cdot \underline{v}_i \qquad \underline{L}_i = \underline{\Theta}_i \cdot \underline{\omega}_i$$

$$\frac{d\underline{p}_i}{dt} = \underline{f}_{\text{ext},i} = m_i \cdot \underline{\dot{v}}_i$$

$$\frac{d\underline{L}_i}{dt} = \underline{\tau}_{\text{ext},i} = \underline{\Theta}_i \cdot \underline{\dot{\omega}}_i + \underline{\omega}_i \times \underline{\Theta}_i \underline{\omega}_i$$

$$\Leftrightarrow$$

Change of velocity

$$\underline{\dot{v}}_i = \frac{1}{m_i} \cdot \underline{f}_{\text{ext},i}$$

$$\underline{\dot{\omega}}_i = \underline{\Theta}_i^{-1} \cdot \left(\underline{\tau}_{\text{ext},i} - \underline{\omega}_i \times \underline{\Theta}_i \underline{\omega}_i\right)$$

Equation of motion of a system of Rigid Bodies

$$\underbrace{\begin{bmatrix} \underline{\dot{v}}_1 \\ \underline{\dot{\omega}}_1 \\ \vdots \\ \underline{\dot{v}}_n \\ \underline{\dot{\omega}}_n \end{bmatrix}}_{\underline{\ddot{x}}} = \underbrace{\begin{bmatrix} m_1^{-1}\underline{I} & \underline{0} & & \underline{0} & \underline{0} \\ \underline{0} & \underline{\Theta}_1^{-1} & & \underline{0} & \underline{0} \\ & & \ddots & & \\ \underline{0} & \underline{0} & & m_n^{-1}\underline{I} & \underline{0} \\ \underline{0} & \underline{0} & & \underline{0} & \underline{\Theta}_n^{-1} \end{bmatrix}}_{\underline{M}^{-1}} \cdot \underbrace{\begin{bmatrix} \underline{f}_{\text{ext},1} \\ \underline{\tau}_{\text{ext},1} - \underline{\omega}_1 \times \underline{\Theta}_1 \underline{\omega}_1 \\ \vdots \\ \underline{f}_{\text{ext},n} \\ \underline{\tau}_{\text{ext},n} - \underline{\omega}_n \times \underline{\Theta}_n \underline{\omega}_n \end{bmatrix}}_{\underline{f}_{\text{ext}}}$$

Linear Momentum : momentum is mass times velocity, and force is the rate of change of momentum, this is just Newton’s Second Law in vector form.

Angular Momentum : torque is the rotational equivalent of force, and angular momentum depends on the inertia tensor and angular velocity, we express it through this derivative.

Derivative

- Forward Dynamics + Integration:

It answers: "**Given the current state and force, what happens next?**"

$$\underline{\ddot{x}} = \underline{M}^{-1} \cdot \underline{f}_{\text{ext}}$$

$$\frac{\underline{\dot{x}}(t+dt) - \underline{\dot{x}}(t)}{dt} = \underline{M}^{-1}(t) \cdot \underline{f}_{\text{ext}}(t)$$

$$\underline{\dot{x}}(t+dt) = \underline{\dot{x}}(t) + dt \cdot \underline{M}^{-1}(t) \cdot \underline{f}_{\text{ext}}(t)$$

if under constraints, Solver finds out what should happen

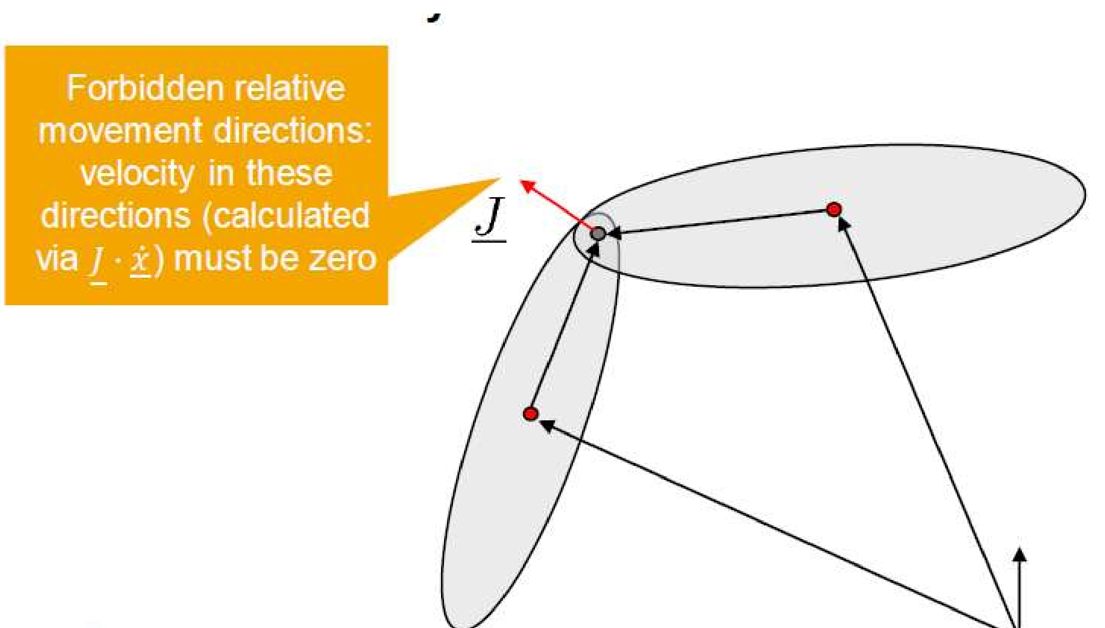
Constraint Jacobian: Describes the “direction of the constraints”:

$$\frac{d}{dt}(C(\underline{x}(t), t)) = \underline{0}$$

$$\frac{\partial C}{\partial \underline{x}}\dot{\underline{x}} + \frac{\partial C}{\partial t} = \underline{0}$$

$$\underline{J} \cdot \dot{\underline{x}} + \underbrace{\frac{\partial C}{\partial t}}_{=\underline{0}} = \underline{0}$$

Lagrange Multipliers: Define the magnitude of the constraint forces



traints”

Principle of D’Alembert Constraint forces fcmust not apply any power Pc

what is b: Non-zero terms, generally used to include drift correction terms or non-homogeneous constraints（As time goes by, constraints may drift due to numerical errors.)

"**J times x-dot not** equals 0.”

$$\underline{J} \cdot \dot{\underline{x}}(t + dt) = \underline{b}$$

3.Properties and Kinematics of Rigid Bodies

ref: C10.c

- **Rigid Bodies**: Non-deformable objects with 6 degrees of freedom (DOF).
- Centre of Mass:where the weighted relative position of the distributed mass vanishes.
- Inertia Tensor: The Inertia Tensor describes the relationship between angular velocity and angular momentum. Symmetric and Positive definite →Eigenvectors of Inertia Tensor
- Parallel Axis Theorem: allows to change the reference point of the Inertia Tensor.

- How do we express the position of the center of mass in the world coordinate system?
  Position and orientation of a Rigid Body
  Orientation → Quaternions （Quaternions are able to represent rotations in three dimensional space）

State Vector → Constraint Formulation → Inverse Dynamics → Forward Dynamics → State Vector

4. Constraint Modeling

ref: C10.d

Constraints are used in mechanical engineering to restrict the degrees of freedom of a dynamics system

- **Holonomic Constraints**: **Equality constraints** on position/orientation, e.g., joints.
- **Non-holonomic Constraints**: Constraints on velocity or **inequality conditions** (e.g., rolling, contact).

Three-level constraint derivation process:

- Position-Based: Enforce the relative position between two anchor points to remain constant.

- Velocity-Based: Ensure that there is no relative velocity between contact points.
- Acceleration-Based

Jacobian Matrix：

The first-order derivative of the constraint equation with respect to the state variable;

Used to transform the constraint into a linear equation;

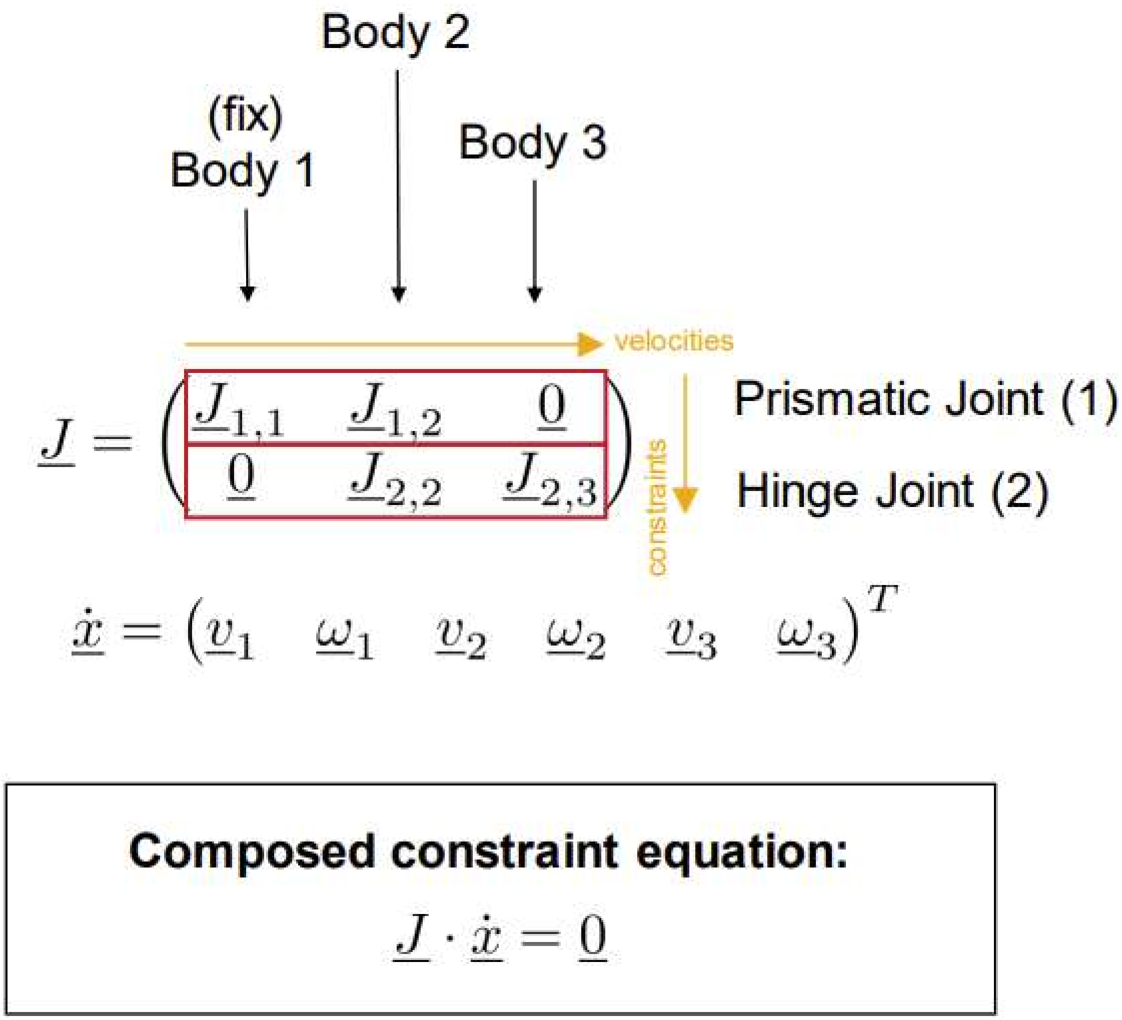
$$\underline{J}(\underline{x}) = \frac{\partial f}{\partial \underline{x}_j}(\underline{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\underline{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\underline{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\underline{x}) & \cdots & \frac{\partial f_m}{\partial x_n}(\underline{x}) \end{pmatrix}$$

Constraint Modeling: World Coordinate System vs. Joint Coordinate System

World Frame: Standard modeling method, easy to unify globally;

Joint Frame: Captures certain constraint directions (such as sliding directions) more clearly, making coding easier;

Combination of multiple constraint systems:



5. Simulation of free-floating Rigid Bodies

ref: C10.e

Conservation of Momentum

Linear Momentum

Angular Momentum

Momentum Change：If an external force or torque is applied, momentum will change with time，If the external force does not act on the center of mass, additional torque will be generated (i.e., the eccentric load causes rotation).

$$\underline{f}_{\text{ext}} = \dot{\underline{p}} = m \cdot \dot{\underline{v}}$$

$$\underline{\tau}_{\text{ext}} = \dot{\underline{L}} = \underline{\Theta} \cdot \dot{\underline{\omega}} + \underline{\omega} \times \underline{\Theta} \cdot \underline{\omega}$$

If the external force attacks off-centre of the centre of gravity it creates an additional torque

$$\underline{\tau}'_{\text{ext}} = \underline{r} \times \underline{f}'_{\text{ext}}$$

Gyroscopic Torque

Semi-Implicit Integration

⚠️ 显式欧拉的问题

它虽然简单，但**数值稳定性差**，容易导致刚体系统震荡、爆炸，尤其在有弹簧、碰撞或高频加速度时。

✅ 二、什么是“半隐式欧拉法”？

半隐式欧拉法（Semi-Implicit Euler，也叫 Symplectic Euler）做了一点关键改进：

✅ 核心思想：

> **先更新速度，再用新的速度去更新位置。**

$$v_{t+\Delta t} = v_t + \Delta t \cdot a_t \quad \text{（先更新速度）}$$
$$x_{t+\Delta t} = x_t + \Delta t \cdot v_{t+\Delta t} \quad \text{（再用新速度更新位置）}$$

Simulation loop：

Current system state → Calculation of derivative of system state→ Recalculation of inertia tensor→ Current system state

6.Constraint forces and simulation

ref: C10.f

D'Alembert Principle:The constraint force does no work

Two methods for calculating restraint force：

- Equation System Approach：
- **JMJT Approach**

JMJT-approach

$$\underline{J}\cdot\underline{M}^{-1}\cdot\underline{J}^T\cdot\underline{\lambda}+\underline{J}\cdot\underline{M}^{-1}\cdot\underline{f}_{ext}-\underline{b}=\underline{0}$$

Equation system approach

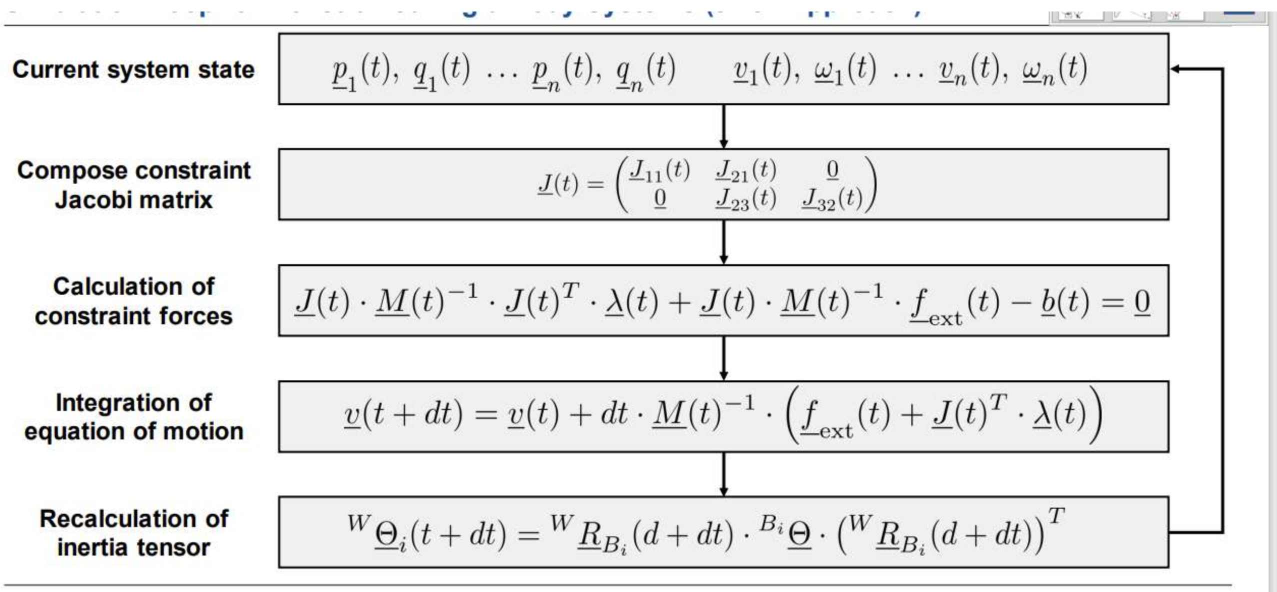
$$\begin{pmatrix}\underline{M} & -\underline{J}^T\\ \underline{J} & \underline{0}\end{pmatrix}\cdot\begin{pmatrix}\ddot{\underline{x}}\\ \underline{\lambda}\end{pmatrix}-\begin{pmatrix}\underline{f}_{ext}\\ \underline{b}\end{pmatrix}=\begin{pmatrix}\underline{0}\\ \underline{0}\end{pmatrix}$$

Advantages of the JMJT method：

- Much smaller equation system. Thus, can be calculated much more efficiently
- Forward dynamics is calculated independent

Simulation Loop for Constrained Rigid Body Systems：

Current system state → Compose constraint Jacobi matrix → Calculation of constraint forces → Integration of equation of motion → Recalculation of inertia tensor



Constraint Drift：Multi-step integration may lead to "loose anchor points" and "broken chains" （Baumgarte）

Generalized Coordinates vs Maximal Coordinates（Prone to drift, requires correction）

7.Stabilization

ref: C10.g

Due to numerical errors and integral drift, the constraints will gradually become "loose" over time.

Add feedback to the original acceleration constraint.

Baumgarte stabilization

$$\ddot{C}(\underline{x}, t) + \alpha \cdot \dot{C}(\underline{x}, t) + \beta \cdot C(\underline{x}, t) = \underline{0}$$

Reformulate depending on constraint Jacobi matrix:

$$\underbrace{\underline{J} \cdot \underline{\ddot{x}} \quad - \quad \underline{b}}_{\text{relative acceleration}} \quad + \quad \alpha \cdot \underbrace{\underline{J} \cdot \underline{\dot{x}}}_{\text{relative velocity}} \quad + \quad \beta \cdot \underbrace{\underline{x}_{\text{rel}}}_{\text{violation of position-based constraint}} = \underline{0}$$

$$\underline{x} = \begin{pmatrix} \underline{x}_1 & \underline{q}_1 & \cdots & \underline{x}_n & \underline{q}_n \end{pmatrix}^T$$

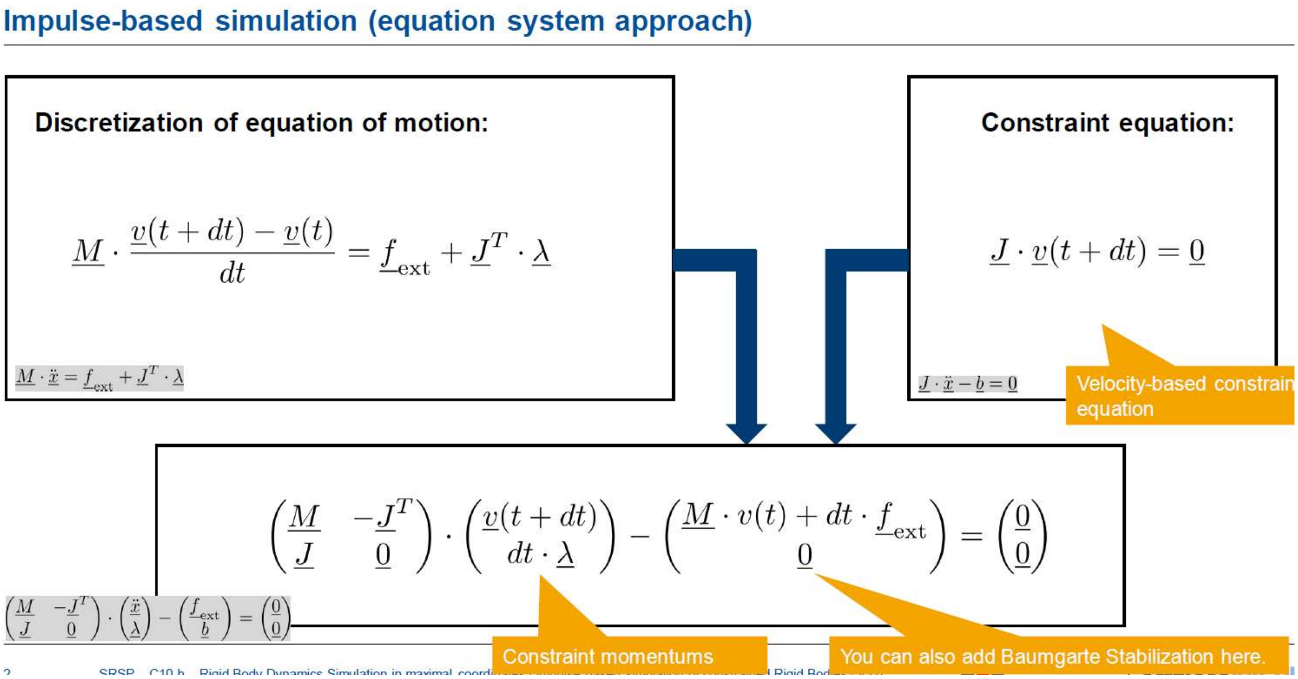
Corresponds to mass-spring-damper oscillator:

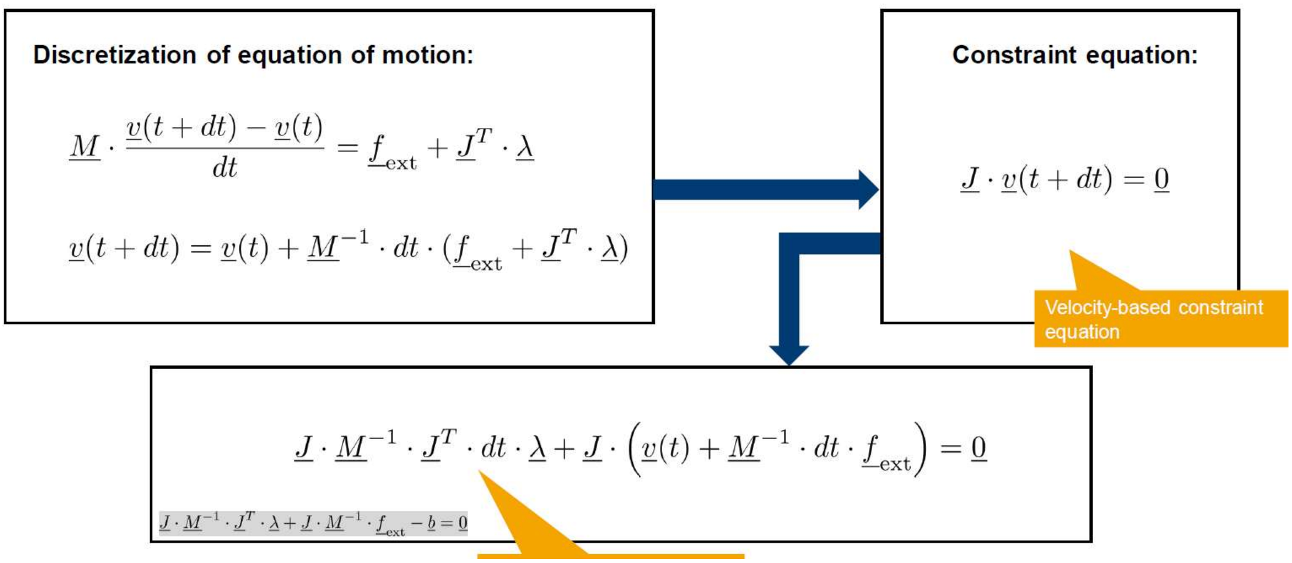
$$m \cdot \underline{\ddot{x}}_{\text{rel}} + d \cdot \underline{\dot{x}}_{\text{rel}} + k \cdot \underline{x}_{\text{rel}} = 0$$

8.Impulse-based simulation of constrained Rigid Bodies

ref: C10.h

This part is important because I am working on a solver for collision detection, so the impact is particularly important.





Velocity changes (impulse),

not directly dealing with acceleration

More suitable for dealing with collision events

- **Velocity-based constraint equations can be used**
- **Larger integration steps possible**： In the traditional acceleration method, if the time step is too large，Numerical oscillation and instability are likely to occur; Stabilization techniques such as Baumgarte must be introduced; The impulse method directly corrects at the velocity level and does not require integrated acceleration; The stability requirements for the time step are more relaxed.

- **Easier integration of friction**： In the acceleration method, friction force must be introduced in a complicated way through differentiation and integration;
- No ambiguities if coulomb friction is modelled between two bodies

If you are simulating a continuous mechanical system (such as robot joints, non-impact system) - it is recommended to use the acceleration method;

If the system frequently experiences contact, impact, and sliding friction - it is recommended to use the impulse method;

When in contact or collision, the velocity changes suddenly in a very short time, and the acceleration tends to infinity (not integrable);

9.Simulation of constrained Rigid Bodies with contacts

ref: C10.i

Next let's talk about how to simulate contact rigid bodies with constraints.

Rigid bodies will not deform, which means that there will be no penetration between them, so this creates a Non-holonomic Constraints, which means an inequality constraint.

Problems with nonholonomic constraints in inverse dynamics

$$
\begin{array}{l}
\\
\textbf{Holonomic constraints:} \\
\textbf{Non-holonomic constraints:}
\end{array}
\begin{pmatrix} \underline{M} & -\underline{J}_e^T & -\underline{J}_n^T \\ \underline{J}_e & \underline{0} & \underline{0} \\ \underline{J}_n & \underline{0} & \underline{0} \end{pmatrix} \cdot \begin{pmatrix} \underline{v}(t+dt) \\ dt \cdot \underline{\lambda}_e \\ dt \cdot \underline{\lambda}_n \end{pmatrix} - \begin{pmatrix} \underline{M} \cdot v(t) + dt \cdot \underline{f}_{\text{ext}} \\ \underline{0} \\ \underline{0} \end{pmatrix} = \begin{pmatrix} \underline{0} \\ \underline{0} \\ \underline{a}_n \end{pmatrix}
$$

$$\underline{a}_n \geq \underline{0}$$

Problems：

- Ambiguity of contact forces

- Realization of resting contact

Complementarity Constraints

Mixed Linear Complementarity Problem

$$\begin{pmatrix} \underline{M} & -\underline{J}_e^T & -\underline{J}_n^T \\ \underline{J}_e & \underline{0} & \underline{0} \\ \underline{J}_n & \underline{0} & \underline{0} \end{pmatrix} \cdot \begin{pmatrix} \underline{v}(t+dt) \\ dt \cdot \underline{\lambda}_e \\ dt \cdot \underline{\lambda}_n \end{pmatrix} - \begin{pmatrix} \underline{M} \cdot v(t) + dt \cdot \underline{f}_{\text{ext}} \\ \underline{0} \\ \underline{0} \end{pmatrix} = \begin{pmatrix} \underline{0} \\ \underline{0} \\ \underline{a}_n \end{pmatrix}$$

$$\underline{a}_n \geq \underline{0} \quad , \quad \underline{\lambda}_n \geq \underline{0} \quad , \quad \underline{a}_n^T \cdot \underline{\lambda}_n = 0$$

Case 1: Bodies are separating

$$\underline{a}_n \geq \underline{0} \quad , \quad \underline{\lambda}_n = \underline{0}$$

Case 2: Bodies are resting on each other

$$\underline{a}_n = \underline{0} \quad , \quad \underline{\lambda}_n \geq \underline{0}$$

Complementarity constraints (JMJT-approach)

Mixed Linear Complementarity Problem

$$\begin{pmatrix} \underline{J}_e \underline{M}^{-1} \underline{J}_e^T & \underline{J}_e \underline{M}^{-1} \underline{J}_n^T \\ \underline{J}_n \underline{M}^{-1} \underline{J}_e^T & \underline{J}_n \underline{M}^{-1} \underline{J}_n^T \end{pmatrix} \cdot \begin{pmatrix} dt \cdot \underline{\lambda}_e \\ dt \cdot \underline{\lambda}_n \end{pmatrix} + \begin{pmatrix} \underline{J}_e \\ \underline{J}_n \end{pmatrix} \cdot \left(\underline{v}(t) + \underline{M}^{-1} dt \cdot \underline{f}_{\text{ext}} \right) = \begin{pmatrix} \underline{0} \\ \underline{a}_n \end{pmatrix}$$

$$\underline{a}_n \geq \underline{0} \quad , \quad \underline{\lambda}_n \geq \underline{0} \quad , \quad \underline{a}_n^T \cdot \underline{\lambda}_n = 0$$

Slack variable: Relative velocity after applying constraint impulses

Case 1: Bodies are separating

$$\underline{a}_n \geq \underline{0} \quad , \quad \underline{\lambda}_n = \underline{0}$$

Case 2: Bodies are resting on each other

$$\underline{a}_n = \underline{0} \quad , \quad \underline{\lambda}_n \geq \underline{0}$$

$$\underline{J} \cdot \underline{M}^{-1} \cdot \underline{J}^T \cdot dt \cdot \underline{\lambda} + \underline{J} \cdot \left(\underline{v}(t) + \underline{M}^{-1} \cdot dt \cdot \underline{f}_{ext} \right) = \underline{0}$$

Projected Gauss-Seidel 解法

Algorithm 1 Projected Gauss-Seidel Algorithm

determine $\underline{A}, \underline{B}, \underline{C}, \underline{D}$
determine $\underline{L}_A, \underline{U}_A, \underline{L}_B, \underline{U}_B$
calculate $\underline{L}_A^{-1}, \underline{L}_B^{-1}$
guess initial $\underline{\lambda}_0$

Could be used for warm start or just use an arbitrary guess

$\underline{\lambda}^k = \underline{\lambda}_0$
error $= \infty$
while $\text{error} > \text{errorLimit}$ **do**

The error criteria could also be replaced by a maximal number of iterations

$\quad \underline{\lambda}_e^{k+1} = \underline{L}_A^{-1} \left(\underline{b}_e + \underline{C} \cdot \underline{\lambda}_n^k - \underline{U}_A \cdot \underline{\lambda}_e^k \right)$
$\quad \underline{\lambda}_n^{k+1} = \underline{L}_B^{-1} \left(\underline{b}_n - \underline{D} \cdot \underline{\lambda}_e^k - \underline{U}_B \cdot \underline{\lambda}_n^k \right)$
$\quad \underline{\lambda}_n^{k+1} = \max(0, \underline{\lambda}_n^{k+1})$

Clamp all complementarity constraints (projection step)

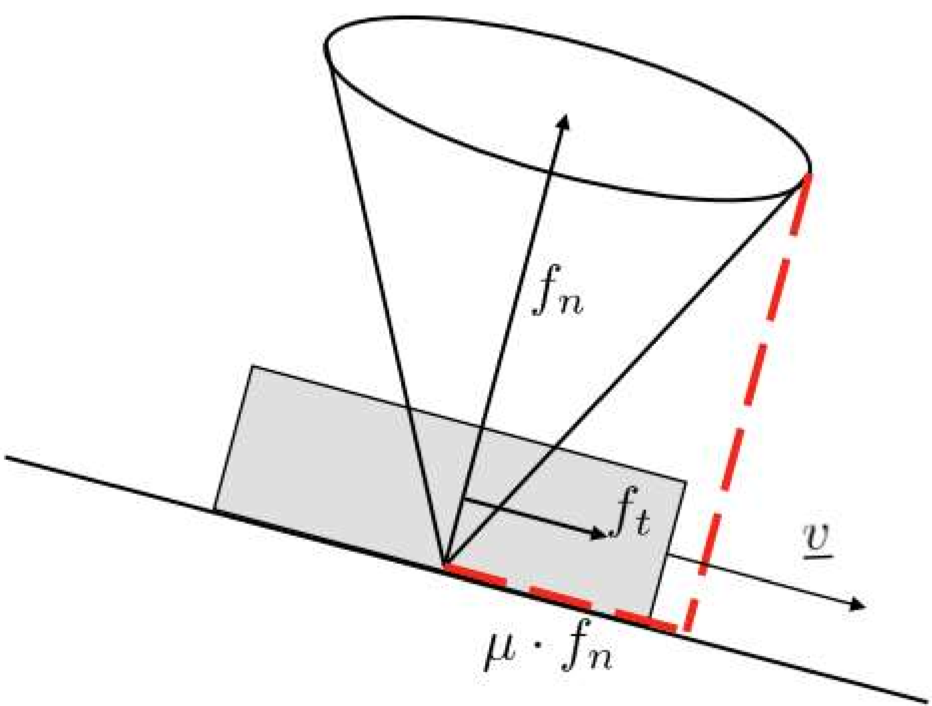
$\quad$ error $= |\underline{\lambda}^{k+1} - \underline{\lambda}^k|$
$\quad \underline{\lambda}_k = \underline{\lambda}_{k+1}$
end while

10.Simulation of Constrained Rigid Bodies with Friction

Coulomb friction model:

$$f_t \leq \mu \cdot f_n$$

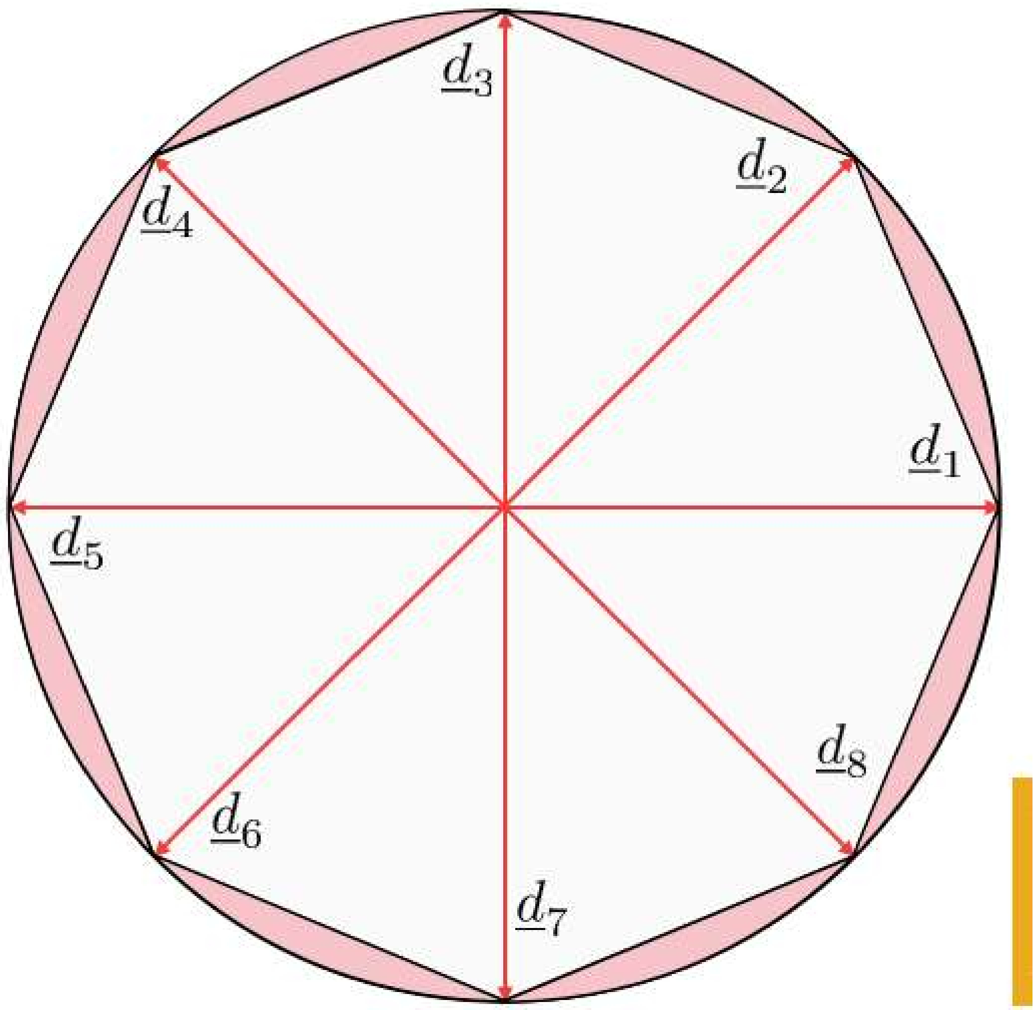
friction cone



Static (sticking to the slope) Friction is inside the cone and is less than the maximum allowable value

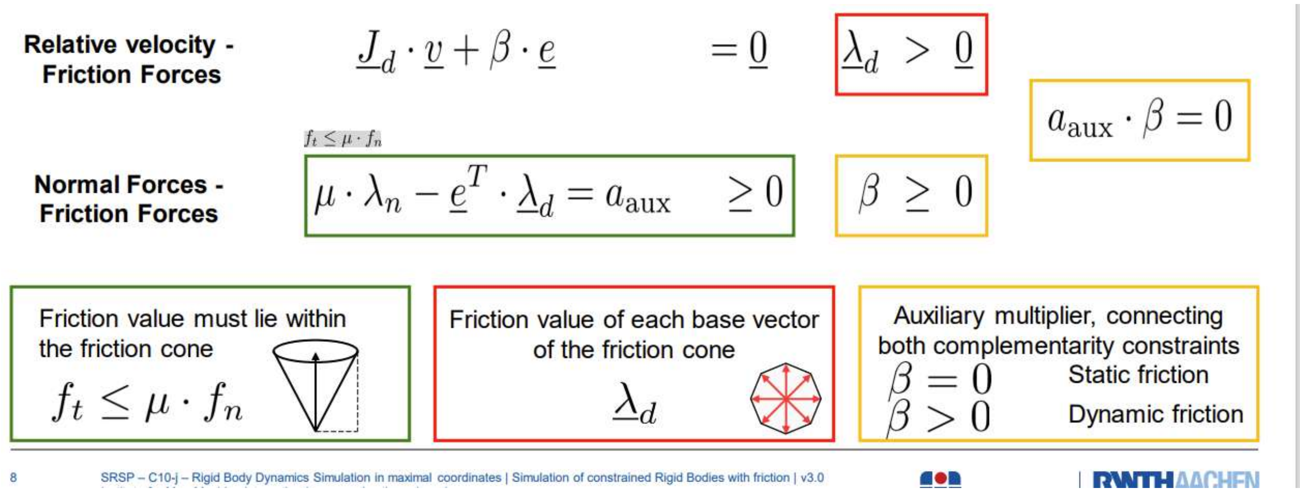
Sliding (sliding down) Friction is on the cone, reaches the maximum value, and the direction resists the sliding direction

Friction Cone Approximation：Friction Pyramid Use multiple orthogonal basis vectors to form a tangent basis; Need to find a balance Accuracy vs. Runtime Trade-off



Friction modeling requires the introduction of two complementary constraints (core)

- Supplementary variables: β auxiliary variable representing sliding velocity
- The friction force cannot exceed the friction cone boundary.



Next let's talk about how to simulate contact rigid bodies with constraints.

Rigid bodies will not deform, which means that there will be no penetration between them, so this creates a Non-holonomic Constraints, which means an inequality constraint.

First, let's look at an example. The first step is to convert the position of the contact point from the rigid body coordinate system to the world coordinate system.

Then we use Non-holonomic Constraints to get the eqution. and Derived twice, we can get Relative velocity in contact normal direction eqution and Relative acceleration in contact normal direction eqution.

magnitude

Substitute the Equation of motion into the Constraint equation

Static and kinetic friction

condition