Session 7: Stability and Dynamics

Part A- Stability
Part B: Zero Moment Point
Part c: Dynamics

Dr. Arshiya Sood

June 12, 2025

Table of Contents

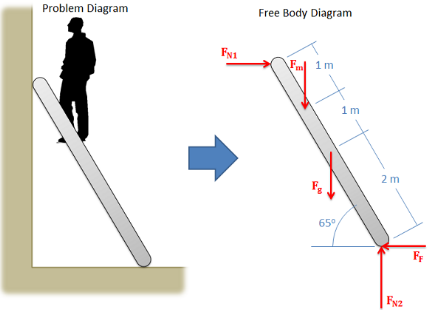
Session 6 Recap & Reflect: What Did We Learn?

Session 6 Recap – Free Body Diagrams (FBD)

- **Free Body Diagram (FBD):** Simplified sketch showing all external forces and torques on a body.
- **Purpose:**
  - Helps isolate and analyze mechanical systems.
  - Essential for applying Newton's Laws to robotics problems.
- **Key Features:**
  - All forces labeled (e.g., F_{grav}, F_{norm}, F_{fric}).
  - Arrows for force direction and magnitude.
  - Moments shown as curved arrows.
  - Clear coordinate system.
- **Application in Robotics:**
  - Calculate joint torques, actuator sizing, balance stability analysis.
  - Helps avoid incorrect assumptions or overlooked forces.

Session 6 Recap – Drawing FBDs

- **Steps to Draw FBDs:**
  1. Isolate the object.
  2. Identify contact surfaces.
  3. Define coordinate system.
  4. Add contact forces (normal, friction, tension, applied).
  5. Add non-contact forces (gravity, electromagnetics if applicable).
  6. Resolve angled forces into components.
  7. For multi-body systems: draw one FBD per object.



- **Support Reactions:**
  - Roller: vertical reaction only.
  - Hinge: horizontal + vertical reactions.
  - Fixed: horizontal, vertical, and moment reactions.

Session 6 Recap – Center of Mass (COM)

- **Center of Mass (COM):** Point representing the average position of mass.
- **Key Properties:**
  - Unique for every system and time instant.
  - Can lie inside or outside the object.
  - Simplifies analysis of complex multi-body systems.
- **COM vs COG:**
  - COM: purely mass distribution.
  - COG: includes gravity

Session 6 Recap – COM Equations and Examples

- **Equations:**
  - Discrete System:

$$\vec{r}_{\text{COM}} = \frac{\sum m_i \vec{r}_i}{\sum m_i}$$

    where
    - m_i is the mass of the i^{th} component
    - $\vec{r}_i$ is the position vector of that component
  - Continuous System:

$$\vec{r}_{\text{COM}} = \frac{1}{M_{\text{tot}}} \int \vec{r}\, dm \quad \text{where} \quad dm = \begin{cases} \rho\, dV & \text{(volume)} \\ \rho\, dA & \text{(area)} \\ \rho\, ds & \text{(curve)} \end{cases}$$

- **Example Highlights:**
  - Two-point masses: COM lies along connecting line, closer to heavier mass.
  - Uniform rod: COM at midpoint.
  - Human body arm movement: raising arms shifts COM upward by 12 cm.

Recap Quiz

Q1. A robotic manipulator lifts a payload of 5 kg at its end-effector. The arm length is 1.2 m and has its own mass of 3 kg uniformly distributed along its length. What is the correct free-body approach to calculate the total torque required at the shoulder joint?

A. Only consider the torque due to payload mass.

B. Add the torque from both payload mass and arm's COM.

C. Use only the arm's total mass multiplied by full arm length.

D. Use only the payload mass multiplied by full arm length.

Q2. In a quadcopter drone, shifting a heavy battery from the center toward one corner affects:

A. The total mass but not the COM.

B. The COM shifts toward the corner and reduces stability.

C. The COM remains at the geometric center due to symmetry.

D. The COM moves upward, improving stability.

Recap Quiz (contd.)

Q3. Which of the following is an incorrect practice while drawing an FBD of a robotic system?

A. Including all external forces.
B. Representing each moment and torque acting.
C. Including internal forces between robot components.
D. Choosing a convenient coordinate system.

Q4. A robotic gripper holds a box weighing 50 N using two parallel fingers that apply equal and opposite normal forces on the sides of the box. What condition ensures the box remains stationary in the gripper?

A. Normal force must exceed the weight of the box.
B. Friction force from both fingers must balance the weight.
C. Sum of friction forces equals or exceeds the weight
D. Weight is irrelevant as only horizontal forces act.

Recap Quiz (contd.)

Q5. A robotic crane lifts a payload of mass m using a cable inclined at an angle θ to the vertical. Which force component is responsible for balancing the horizontal motion tendency of the payload?

A. Vertical component of tension.

B. Gravitational force mg.

C. Horizontal component of tension.

D. Friction at the crane pulley.

Q6. A robot places an object on an inclined plane at 30°. While drawing the FBD of the object, which component of gravity acts along the plane?

A. mg

B. $mg \cdot \sin(30^\circ)$

C. $mg \cdot \cos(30^\circ)$

D. Zero, as object is stationary.

Recap Quiz (contd.)

Q7. A mobile robot climbs a hill while tethered with a cable providing support. Which force will contribute to both pulling force and frictional drag in the FBD?

A. The gravitational normal force.
B. The component of gravity parallel to incline.
C. The tension force in the tether.
D. The rolling resistance force.

Q8. As a robotic manipulator extends its arm forward while holding a payload:

A. COM of the robot shifts forward, increasing stability.
B. COM shifts backward, reducing tipping risk.
C. COM shifts forward, increasing tipping risk.
D. COM remains unchanged due to fixed base.

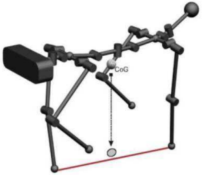
Answer Key with Explanations

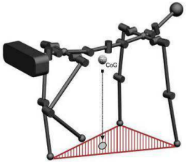
- **Q1: B** Both payload and arm mass contribute to torque; arm's mass acts at its COM (midpoint).
- **Q2: B** Shifting battery shifts COM toward corner, increasing tipping risk and reducing stability.
- **Q3: C** Internal forces between robot components should not be included in FBD; only external forces are shown.
- **Q4: C** Friction must balance weight; friction depends on normal force via $F_{fric} = \mu \cdot F_{norm}$.
- **Q5: C** The horizontal component of tension balances horizontal motion tendency when cable is inclined.
- **Q6: B** Gravity along incline is $mg \cdot \sin(30^\circ)$; perpendicular component is $mg \cdot \cos(30^\circ)$.
- **Q7: C** The tether tension contributes both pulling force and frictional drag along incline.
- **Q8: C** As arm extends, COM shifts forward, increasing tipping moment arm, reducing stability.

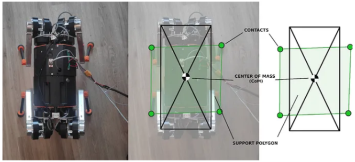
Part A: Support Polygon and stability

Support Polygon: Definition

- In robotics, the **Support Polygon (SP)** is a 2D area formed by projecting all ground contact points (feet, wheels, legs) onto a horizontal plane.
- The SP is crucial for analyzing **static stability** — the robot's ability to remain upright under gravity.
- Shape of the support polygon depends on:
  - Number of contact points
  - Arrangement of support points
- Examples:
  - 2-legged robot → line
  - 3-legged robot → triangle
  - 4-legged robot → quadrilateral







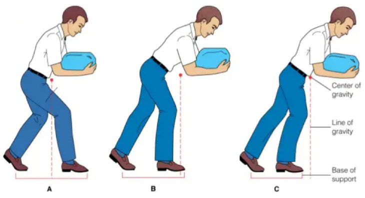
Support polygon representation while standing still

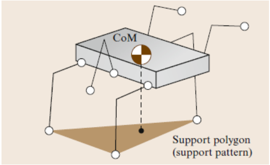
Why is Support Polygon Important?

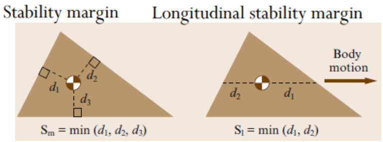
- The **Center of Gravity (COG)** must lie within the support polygon to maintain static stability.
- **Stability Condition:**

$$\begin{cases} \text{COG inside SP} \rightarrow \text{Stable} \\ \text{COG outside SP} \rightarrow \text{Tipping risk} \end{cases}$$

- The **stability margin** is the distance from COG projection to the polygon boundary:
  - Larger margin → higher stability.
  - Smaller margin → higher risk of tipping.
- **Control strategies:** adjust leg/wheel positions to modify the support polygon dynamically and maintain balance.





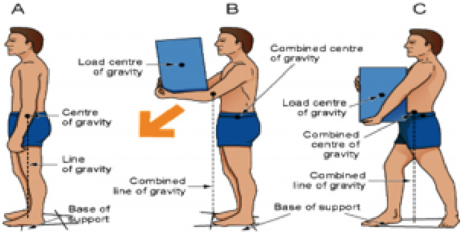


Types of Stability: Static vs Dynamic

- **Static Stability:**
  - Robot can remain upright while stationary without any control input.
  - Achieved by keeping COG within the support polygon.
  - More support points (legs) make static stability easier.
- **Dynamic Stability:**
  - Balance maintained during motion using active control.
  - COG may temporarily move outside support polygon.
  - Requires continuous sensing, feedback, and control.
  - Examples: humans walking, hopping robots, bicycles.

Example: Human Stability and Balance

- **Human balance requires:**
  - Coordinated muscle activity
  - Neurological control
  - Proper skeletal alignment
- **Body Alignment:**
  - Aligns body parts vertically to reduce strain on muscles, ligaments, and joints.
  - Correct posture reduces fall risk.



Stability Principle:

- Vertical line from COG must fall inside Base of Support (BOS) for stability.
- Lowering COG (e.g., bending knees) increases stability.
- Widening stance increases BOS and stability margin.

Key Point: Same principles are applied in humanoid robot balance control.

Legged Locomotion: Stability Challenge

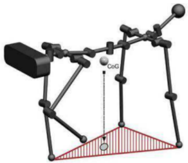
- While most animals use legs for locomotion, legged locomotion is a very difficult problem in robotics.
- Any robot needs to maintain **stability**:
  - Not wobble or fall over easily.
  - Handle uneven terrain and disturbances.
- **Two kinds of stability:**
  - **Static Stability:** Stay upright without any motion or control input.
  - **Dynamic Stability:** Maintain balance during motion using active control.
- Humans appear statically stable but actually use:
  - Muscles, tendons, and nerves for active balance control.
  - Balance is largely unconscious but must be learned.
  - Babies take time to learn standing due to this control complexity.

Support Polygon in Legged Robots

Stability in legged robots depends on whether the **Center of Gravity (COG)** projection stays inside the **Support Polygon**.

- **2-legged robots:**
  - Support polygon reduces to a line segment between two feet.
  - COG alignment on a line is very difficult — requires constant balancing.
  - Essentially unstable without dynamic balance control.

- **3-legged robots:**
  - Support polygon becomes a triangle.
  - Statically stable — COG can remain within triangle even without motion.
  - Tripod robots (e.g., Robix) demonstrate stable standing.



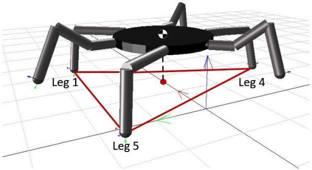
- Adding more legs further enlarges the polygon, improving static stability.

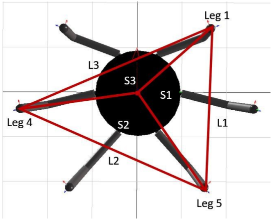
Static Stability and Walking in Legged Robots

- **Lifting a leg affects stability:**
  - When a leg is lifted to step forward, the number of ground contact points reduces.
  - The support polygon reshapes dynamically as legs are lifted and placed.
- **Statically stable walking:**
  - Designed to always keep COG projection inside the current support polygon during the entire gait cycle.
  - Ensures no tipping during leg swing.
- **Minimum number of legs for static walking:**
  - With 3 legs, lifting one leaves only 2 contact points no static stability.
  - At least 4 legs are required to allow for continuous statically stable walking.
- **Tradeoff:**
  - Static walking is very safe but slow and energetically inefficient.

Six-Legged Robots and the Tripod Gait

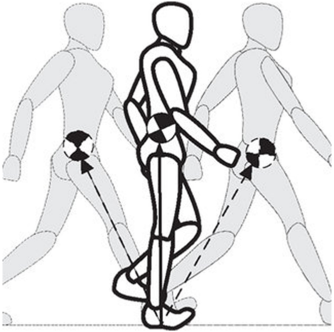
- **Why 6 legs?**
  - 6-legged robots are very popular for achieving high static stability while walking.
  - They allow for statically stable walking even while moving.
- **Tripod Gait:**
  - 3 legs always remain on the ground to form a stable support triangle.
  - The other 3 legs are lifted and moved forward.
  - This process alternates between leg groups.
- **Gait Variants:**
  - **Alternating Tripod Gait:** fixed leg groups alternate.
  - **Ripple Gait:** legs swing sequentially in a wave-like pattern.





Dynamic Stability in Legged Robots

- **Limitation of static stability:**
  - Statically stable walking is very safe but slow.
  - It is also energy inefficient due to constant shifting of legs to maintain COG inside SP.
- **Dynamic Stability:**
  - Allows robots to maintain balance while moving quickly.
  - Center of Gravity (COG) may temporarily move outside the support polygon.
  - Balance is maintained by active control mechanisms.
- **Key Challenge:**
  - Requires continuous sensing, feedback, and control (inverse pendulum problem).
  - Much harder to implement but enables faster, more natural gaits.



Part B: Zero Moment Point (ZMP)

Why ZMP Is Needed?

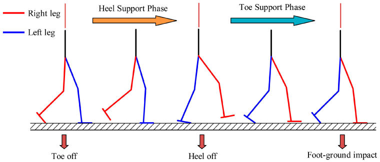
- Humanoid Walking must continuously control:
  - Center of Mass (CM) position, velocity, and acceleration
  - Motion of feet for stepping
  - Motion of hands for manipulation
- The environment heavily influences motion:
  - Obstacles
  - Force disturbances
  - Terrain conditions
- **Unique problem for humanoid robots:**
  - All joints are powered **except** ground contact.
  - Ground contact is not actuated $\rightarrow$ robot is under-actuated at contact point.
  - Loss of balance possible due to external disturbance or poor control.
- **ZMP provides a dynamic balancing condition to prevent falling.**

The Ground Contact Problem in Humanoid Walking

Key control objective:

- Ensure correct contact of feet with ground.
- Prevent unwanted rotations or loss of contact during walking.

- **In lateral direction:**
  - Prevent rolling of the foot on its edge.
  - Avoid ankle twisting by maintaining full foot contact sideways.
- **In forward direction:**
  - **While standing:** entire foot flat on ground.
  - **While walking:**
    - Early stance phase: full foot flat on ground.
    - Late stance phase: heel lifts; ball of foot maintains contact.



The robot must actively plan its motions to respect these contact conditions throughout walking cycles.

What is ZMP? — Definition and Interpretation

Definition: Point on the ground where the sum of horizontal moments due to ground reaction forces equals zero.

- Condition:

$$\sum M_x = 0, \quad \sum M_y = 0$$

- Coordinate frame:
  - x, y: tangential to ground
  - z: vertical (normal to ground)

Physical Interpretation:

- When ZMP lies inside the support polygon:
  - Foot remains flat.
  - No rotational tipping about edges.

Alternate View (Internal Forces):

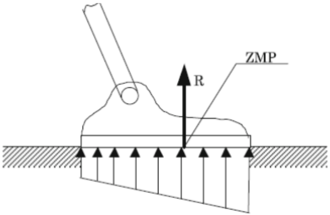
- ZMP can also be seen as the point where:
  - Gravitational forces
  - Inertial forces (from robot motion)
  - External forces

  - produce no horizontal moment.

- ZMP reflects **dynamic balance**:
  - Even if CoM projection moves outside SP during motion,
  - ZMP must stay inside SP for stable contact.

ZMP Equation: Mathematical Formulation

- ZMP location depends on:
  - Center of Mass (CoM) position
  - CoM acceleration
  - Net moment about the CoM, and Gravity



- **ZMP equations:**

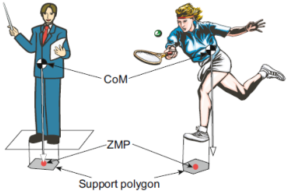
$$x_{ZMP} = x_{CM} - \frac{z_{CM}}{Mg}\left(\ddot{x}_{CM} + \frac{M_y}{M z_{CM}}\right)$$

$$y_{ZMP} = y_{CM} - \frac{z_{CM}}{Mg}\left(\ddot{y}_{CM} - \frac{M_x}{M z_{CM}}\right)$$

- Where:
  - M = robot mass
  - (x_{CM}, y_{CM}, z_{CM}) = CoM position
  - $\ddot{x}_{CM}, \ddot{y}_{CM}$ = CoM accelerations
  - M_x, M_y = net moments about CoM, and g = gravitational acceleration

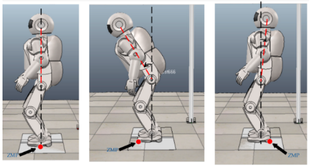
Difference Between CoM and ZMP (Conceptual View)

- **During static standing:**
  - CoM projection coincides with ZMP.
  - Both lie inside support polygon.
- **During dynamic motion:**
  - CoM projection may temporarily leave the support polygon.
  - ZMP must stay inside support polygon to maintain balance.
- **Why ZMP is better for control:**
  - Directly reflects dynamic balance.
  - Used in humanoid walking algorithms.



(a)A standing human (b)A human in action

Figure 12. Zero moment point (ZMP) movement in the x-direction: (**a**) robot standing upright, (**b**) center of gravity of the robot moving forward and (**c**) center of gravity of the robot moving back.



(a) (b) (c)

- ZMP is the key indicator for stability control during walking.
- Humanoid controllers monitor ZMP trajectory to prevent falls.

Practical Examples: CoG, Support Polygon, and ZMP

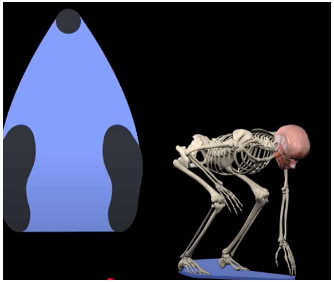
Video Source:
https://www.youtube.com/watch?v=5s1gEi5wfLg

Key Concept Recap:

- BOS: the area beneath a person covering all contact points with the ground.
- CoG: the point where body weight can be considered to act.
- Stability improves when CoG projection falls inside the BOS.

Video Explanation:

- Demonstrates how shifting posture and foot placement alters the BOS.
- Shows CoG projection moving relative to BOS.
- Highlights the necessity of keeping CoG within the support base for balance.



Support Polygon Visualization: Static Balance Test

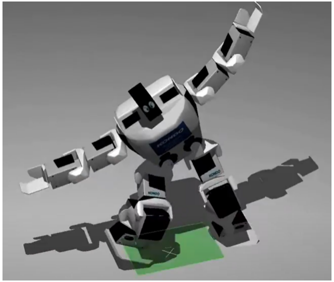
Video Source:
https://www.youtube.com/watch?v=R-dWaxsuWDc

Key Concept Recap:

- Support polygon: 2D projection of ground contact points.
- Stability condition:
  - CoM inside polygon → stable, CoM outside polygon → unstable.

Video Explanation:

- CoM shown as cross (+).
- Support polygon displayed dynamically.
- Polygon color:
  - Green: stable,
  - Red: unstable.
- Limb movement modifies polygon shape.



ZMP Control in Dynamic Walking: Humanoid Robot Example

Video Source:

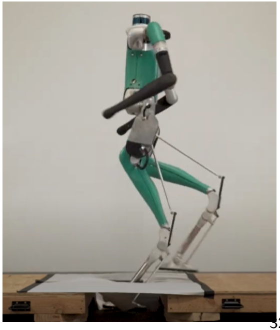
https://www.youtube.com/watch?v=mpxrnrR$_T sg$

Key Concept Recap:

- The point where net moment about horizontal axes equals zero.
- ZMP must remain inside support polygon to prevent falling.
- ZMP shifts dynamically as robot moves.

Video Explanation:

- Humanoid robot adapts walking pattern on unstable terrain.
- CoM moves forward, but ZMP remains inside support polygon.
- Active control of foot placement, posture, and joints maintains stability.
- Terrain disturbances (rocks, gaps) force continuous ZMP adjustments to maintain balance.



Newton-Euler Equation.

Newton-Euler Formulation: Core Idea

The Newton-Euler formulation describes the motion of a rigid body by combining translational and rotational dynamics.

Translational (Linear) Motion — Newton's 2nd Law:

$$\mathbf{F} = m\mathbf{a}_C$$

where $\mathbf{F}$ is the net external force, m is the mass, and $\mathbf{a}_C$ is the acceleration of the center of mass.

Rotational (Angular) Motion — Euler's Equation:

$$\boldsymbol{\tau} = I\boldsymbol{\alpha} + \boldsymbol{\omega} \times (I\boldsymbol{\omega})$$

where $\boldsymbol{\tau}$ is the net external torque, I is the inertia tensor about the CoM, $\boldsymbol{\alpha}$ is the angular acceleration, and $\boldsymbol{\omega}$ is the angular velocity.

Newton-Euler Formulation: Core Idea (contd.)

These two equations together fully describe rigid body motion in 3D.

The Newton-Euler formulation is the foundation for:

- Robot link dynamics
- Inverse dynamics
- Control design
- Whole-body dynamic models

Equations of Motion: General Form

The equations of motion of a physical system describe its **motion** as a function of **time** and optional **control inputs**. In their general form, they are written:

$$F(q(t), \dot{q}(t), \ddot{q}(t), u(t), t) = 0$$

Where:

- t: the time variable,
- q: the vector of generalized coordinates (e.g., joint angles for a manipulator),
- $\dot{q}$: first time-derivative (velocity) of q,
- $\ddot{q}$: second time-derivative (acceleration) of q,
- u: vector of control inputs.

These equations provide a mapping between the **control space** (the actuator commands) and the **state space** of robot motions.

Example: Block Sliding on a Table

Imagine a rigid block sliding on a table, which we can see as a **1-DOF (one degree of freedom)** system with coordinate x. The operator applies a horizontal force u that pushes the block forward.

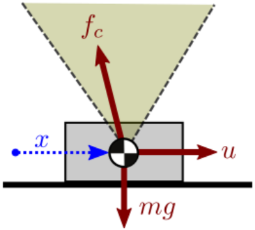
If the force is high enough to overcome friction, applying Newton's second law of motion (and Coulomb's model of sliding friction), we get:

$$m\ddot{x} = u - \mu_k mg$$

where μ_k is the **kinetic coefficient of friction**.

This expression is of the form:

$$F(x, \dot{x}, \ddot{x}, u, t) = 0$$



Newton-Euler Equations

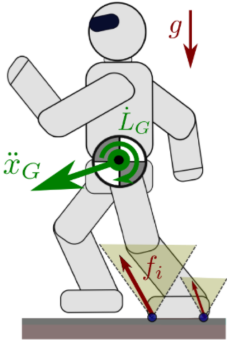
The Newton-Euler equations of motion correspond to the six unactuated coordinates in the equations of motion of our robots.

Newton's equation applies to (linear) translational motions:

$$\sum_{\text{link } i} m_i \ddot{p}_i = mg + \sum_{\text{contact } i} f_i$$

Euler's equation applies to angular motions:

$$\sum_{\text{link } i} (p_i - p_G) \times m_i \ddot{p}_i + I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) = \sum_{\text{contact } i} (p_i - p_G) \times f_i + \tau_i$$



Newton's Equation: Center of Mass

Let p_i denote the position in the world frame of the center of mass of the robot's i^{th} link. Let m_i denote the mass of link i, and $m = \sum_i m_i$ the total mass of the robot. The overall center of mass G is located at the position p_G in the world frame such that:

$$m p_G = \sum_{\text{link } i} m_i p_i$$

In other words, the robot's center of mass is a convex combination of the centers of mass of its links, weighted by their masses.

FORCES

The robot is subject to gravity and contact forces. Let g denote the gravity vector. For a link i in contact with the environment, write f_i as the resultant force exerted on the link.

Newton's Equation: Forces and Internal Forces

Let h_{ij} denote the internal force exerted by link i on link j.
We take $h_{ij} = 0$ if links i and j are not connected (similarly, $f_i = 0$ if link i is not in contact). All force vectors are expressed in the world frame.

Newton's equation of motion links the resultant accelerations and forces:

$$\sum_{\text{link } i} m_i \ddot{p}_i = \sum_{\text{link } i} m_i g + f_i + \sum_{j \neq i} h_{ij}$$

By Newton's third law:

$$h_{ij} = -h_{ji} \implies \sum_i \sum_{j \neq i} h_{ij} = 0$$

Thus:

$$\sum_{\text{link } i} m_i \ddot{p}_i = \sum_{\text{link } i} m_i g + f_i$$

Momentum Form of Newton's Equation

The linear momentum of the robot is defined by:

$$P := m\dot{p}_G$$

Then, Newton's equation can be written as:

$$\dot{P} = m\ddot{p}_G = mg + \sum_{\text{contact } i} f_i$$

In other words, the rate of change of linear momentum equals the sum of external forces exerted on the robot.

Euler's Equation: Setup and Notation

Newton's equation is related to translational motions of the robot. Euler's equation provides a similar relation for angular motions.

Let R_i denote the rotation matrix from the i^{th} link frame to the world frame.

Let ω_i denote the spatial angular velocity of the link, that is, the angular velocity from the link frame to the world frame, expressed in the world frame.

Let I_i denote the inertia matrix of the i^{th} link, expressed in the world frame and taken at the center of mass p_i of the link.

Euler's Equation of Motion

For a link i in contact with the environment, we write τ_i as the resultant moment of contact forces exerted on the link at p_i. If the link is in point contact at p_i, the moment will be zero. In surface contact, both f_i and τ_i may be non-zero.

Euler's equation of motion links angular momentum and external moments:

$$\sum_{\text{link } i} (p_i - p_G) \times m_i \ddot{p}_i + I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) = \sum_{\text{link } i} (p_i - p_G) \times (f_i + m_i g) + \tau_i$$

Because:

$$\sum_{\text{link } i} (p_i - p_G) \times m_i g = 0$$

The simplified equation becomes:

$$\sum_{\text{link } i} (p_i - p_G) \times m_i \ddot{p}_i + I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) = \sum_{\text{contact } i} (p_i - p_G) \times f_i + \tau_i$$

Momentum Form of Euler's Equation

The angular momentum of the robot, taken at the center of mass G, is defined by:

$$L_G := \sum_{\text{link } i} (p_i - p_G) \times m_i \dot{p}_i + I_i \omega_i$$

Then, Euler's equation can be written in concise form as:

$$\dot{L}_G = \sum_{\text{contact } i} (p_i - p_G) \times f_i + \tau_i$$

In other words, the rate of change of the angular momentum equals the resultant moment of external forces exerted on the robot.