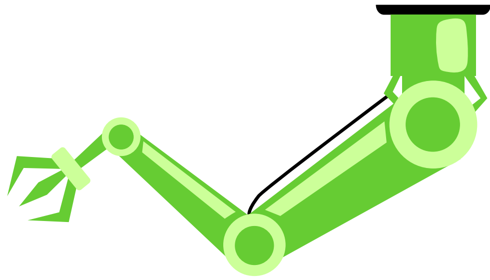
Robot Mechanics



→Lec. 1: Introduction to Robotics

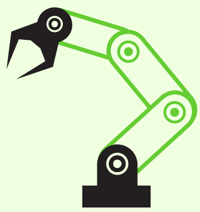
⁕ A **robot** is a programmable machine with actuators, sensors, and control logic, capable of performing complex tasks automatically or semi-autonomously.

Mechanical Manipulator

Kinematic chain: An assembly of **links** connected by **joints** that is the mathematical model for a mechanical system or a machine.

Link: A **rigid body** in a **kinematic chain** that does not deform under motion.

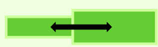
- **End-Effector**: The last link that interacts with the environment.
- **Base**: The fixed link to which all other links are connected.



Joint: A **kinematic pair** which is a connection between two **links** that imposes constraints on their relative movement.

- **Revolute (R) joint**: Links rotate around a common axis.

- **Prismatic (P) joint**: Links slide along a common axis.



Mechanical Manipulator: An industrial **robot** modeled as an actuated **kinematic chain** of links connected by joints in a serial (**open**) arrangement—without forming **closed** loops— designed to control the pose of its final link (**end-effector**) for interaction with the environment.

⁕ **Generalized coordinates** are the minimal set of independent parameters that uniquely specify the position of every part of a system. Constraints reduce the number of these parameters. Here, we consider only **holonomic constraints**, which are expressed as algebraic equations involving coordinates and time.

Degrees of Freedom

The **degrees of freedom (DoF)** of a mechanical system is the number of **generalized coordinates** required to uniquely specify its state. A **manipulator** with $n+1$ links (one fixed as the base), where each rigid link consists of many particles (each needing 3 coordinates, i.e., $3N$ DoF), has only 6 DoF per link due to rigidity. With n revolute or prismatic joints (each removing 5 DoF), the total DoF is:

$$\text{DoF} = 6(n+1) - 6 - 5n = n$$

→Lec. 2: Homogeneous Transformations

1 Frames

✻ In robotics, a frame (or reference frame) is a coordinate system used to define the position and orientation of objects, robots, or their components in physical 3D space. A frame $\{A\}$ can be represented as $\{A\} = \{\mathbf{o}_A, \mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A\}$ where $\mathbf{o}_A \in \mathbb{R}^3$ is the origin of the frame, and $\mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A \in \mathbb{R}^3$ are orthonormal vectors defining the frame axes.

2 Rigid Transformations

✻ Robot links are rigid bodies, so their relative motion can be described only by rigid transformations — combinations of rotations and translations that preserve distances, angles, and handedness (the orientation of the coordinate axes) of the coordinate system. The right-hand rule is a convention and a mnemonic, utilized to define the orientation of axes in 3D space. Coordinates of a point $\mathbf{p}$ expressed in frame B transform to frame A by:

$$^A\mathbf{p} = {}^A\mathbf{R}_B\, {}^B\mathbf{p} + {}^A\mathbf{t}_B$$

Rotation Matrix

The rotation matrix is formed by expressing the unit basis vectors of frame B in frame A:

$$^A\mathbf{R}_B = [{}^A\mathbf{x}_B \quad {}^A\mathbf{y}_B \quad {}^A\mathbf{z}_B]$$

The rotatoin matrix is a proper orthogonal matrix:

$$^A\mathbf{R}_B^\top {}^A\mathbf{R}_B = \mathbf{I}, \quad \det({}^A\mathbf{R}_B) = +1$$

Translation Vector

Positoin vector from origin of frame A to origin of frame B:

$$^A\mathbf{t}_B = {}^A\mathbf{o}_B$$

3 Homogeneous Coordinates

✻ Translation is an affine nonlinear transfomation in 3D Cartesian space but becomes a linear shear in projective space by representing a 3-vector as a 4-vector in homogeneous coordinates. This lets us express any rigid transformation linearly using homogeneous transformation matrices and points.

$$^A\mathbf{p}_h = \begin{bmatrix} ^Ap_x \\ ^Ap_y \\ ^Ap_z \\ 1 \end{bmatrix}$$

Homogeneous Point

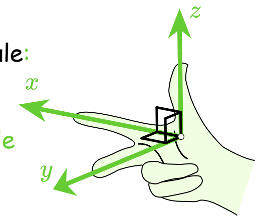
$$^A\mathbf{T}_B = \begin{bmatrix} ^A\mathbf{R}_B & ^A\mathbf{t}_B \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

$$^A\mathbf{p}_h = {}^A\mathbf{T}_B\, {}^B\mathbf{p}_h$$

Homogeneous Transform

✻ Note:

The right-hand rule: point your index finger along the x-axis, your middle finger along the y-axis, and your thumb points along the z-axis.



How to find the inverse transform?

$$^B\mathbf{T}_A = ({}^A\mathbf{T}_B)^{-1}$$

$$^A\mathbf{p} = {}^A\mathbf{R}_B\, {}^B\mathbf{p} + {}^A\mathbf{t}$$

$$^B\mathbf{p} = {}^A\mathbf{R}_B^\top {}^A\mathbf{p} - {}^A\mathbf{R}_B^\top {}^A\mathbf{t}_B$$

$$^B\mathbf{p} = {}^B\mathbf{R}_A\, {}^B\mathbf{p} + {}^B\mathbf{t}_A$$

$$^B\mathbf{T}_A = \begin{bmatrix} ^A\mathbf{R}_B^\top & -{}^A\mathbf{R}_B^\top {}^A\mathbf{t}_B \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

→Lec. 3: Forward Kinematics $\mathbb{x} = f(\mathbb{q})$

Joint Space

The **joint space** (or **configuration space**) $\mathcal{Q}$ is the set of all **joint vectors** that define the robot's pose. Each **joint variable** q_i, called a generalized coordinate, is either a rotation angle θ_i for revolute joints or a linear displacement d_i for prismatic joints. The **joint vector** of all n generalized coordinates $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$ fully specifies the robot's configuration and lies in the n-dimensional joint space $\mathcal{Q} \subseteq \mathbb{R}^n$ $\quad \dim(\mathcal{Q}) = n = \mathrm{DoF}$

Task Space

The **task space** $\mathcal{X}$ (or **cartesian space**/**operational space**) is the space of all possible end-effector poses $\mathbf{x}$ represented by elements of the **Special Euclidean group SE(3)** (the set of all rigid transformations):

$$SE(3) = \left\{\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} : \mathbf{R} \in \{\mathbf{R} \in \mathbb{R}^{3\times3} \mid \mathbf{R}^T\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}, \mathbf{t} \in \mathbb{R}^3\right\}$$

Thus, the maximum dimension of the task space for a rigid end-effector is 6, consisting of 3 positional and 3 orientational degrees of freedom.

✳ The **Denavit-Hartenberg (DH) convention** assigns coordinate frames to robot links to simplify 3D rigid transformations using only **4 parameters** per link instead of 6 parameters. Each transformation is built from four basic steps:

① Rotate about x_i by **link twist** α_i (constant)

② Translate along x_i by **link length** a_i (constant)

③ Translate along z_{i-1} by **link offset** d_i (variable if prismatic, else constant)

④ Rotate about z_{i-1} by **joint angle** θ_i (variable if revolute, else constant)

$${}^{i-1}\mathbf{T}_i = \mathbf{R}_z(\theta_i)\mathbf{T}_z(d_i)\mathbf{T}_x(a_i)\mathbf{R}_x(\alpha_i)$$

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward Kinematics

The **forward kinematics map** $f: \mathcal{Q} \to \mathcal{X}$, $\mathbf{q} \mapsto \mathbf{T}_n^0(\mathbf{q})$ relates the **joint configuration vector** to the **end-effector pose**. The **base frame** $\{0\}$ is fixed to the robot's base or reference point. The **end-effector frame** $\{n\}$ is attached to the robot's tool or terminal link. The overall transformation which defines the end-effector pose with respect to the base frame is given by the product of transformations from each intermediate link frame:

$$\mathbf{T}_n^0(\mathbf{q}) = \prod_{i=1}^{n} \mathbf{T}_i^{i-1}(q_i) = \begin{bmatrix} \mathbf{R}_n^0(\mathbf{q}) & \mathbf{t}_n^0(\mathbf{q}) \\ \mathbf{0}^T & 1 \end{bmatrix}$$

→Lec. 4: Inverse Kinematics $\mathbf{q} = f^{-1}(\mathbf{x})$

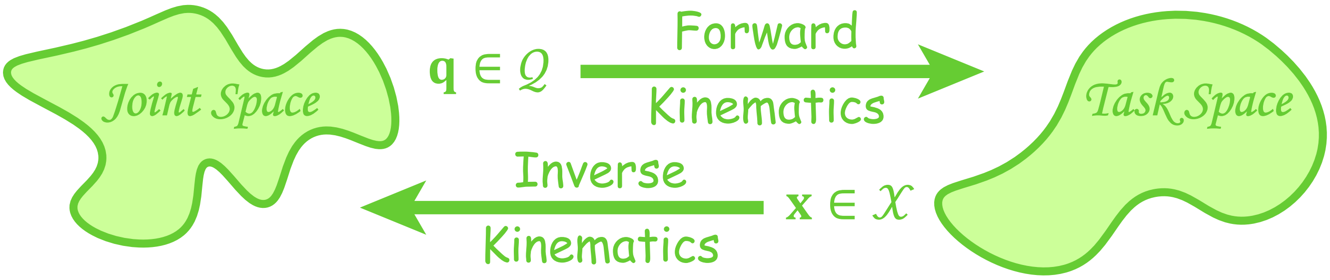
Inverse Kinematics

Given an end-effector pose $\mathbf{x} \in \mathcal{X}$, the inverse kinematics problem asks:

Find joint configuration(s) $\mathbf{q} \in \mathcal{Q}$ "such that" $f(\mathbf{q}) = \mathbf{x}$

If the forward kinematics function $f: \mathcal{Q} \to \mathcal{X}$ is bijective, then a true inverse function $f^{-1}: \mathcal{X} \to \mathcal{Q}$ exists, and each pose $\mathbf{x}$ corresponds to exactly one configuration $\mathbf{q}$. If it is not invertible, we interpret the inverse kinematics solution as a set-valued map: $f^{-1}(\mathbf{x}) := \{\mathbf{q} \in \mathcal{Q} \mid f(\mathbf{q}) = \mathbf{x}\}$, which may be: Empty (no solution), Finite (distinct solutions), Infinite (a solution manifold). The map is said to be locally invertible at a configuration $\mathbf{q}$ if its differential (Jacobian matrix) has full rank. This implies that near $\mathbf{q}$, the mapping behaves like a local diffeomorphism, and small changes in correspond to unique, smooth changes in $\mathbf{x}$. A singularity $\mathbf{q}_s$ is a configuration where the differential is not full rank. At singularities:

The inverse becomes locally non-unique or unstable.
The robot loses one or more instantaneous degrees of freedom.
Control and motion near singularities become problematic and sensitive.



Workspace

Reachable Workspace $\mathcal{W}_R \coloneqq \mathrm{Im}(f) = \{\mathbf{x} \in \mathcal{X} \mid \exists\, \mathbf{q} \in \mathcal{Q} \text{ such that } f(\mathbf{q}) = \mathbf{x}\}$

This is the set of all end-effector poses (position and orientation) that the robot can reach through some joint configuration. If $\mathbf{x} \notin \mathcal{W}_R$, then no inverse kinematics solution exists — the pose is physically unreachable.

Dextrous Workspace $\mathcal{W}_D := \{\mathbf{p} \in \mathbb{R}^3 \mid \forall\, \mathbf{R} \in SO(3),\ \exists\, \mathbf{q} \in \mathcal{Q} \ : f(\mathbf{q}) = (\mathbf{p}, \mathbf{R})\}$

This is the set of positions where the robot can reach every possible orientation of the end-effector. For every $\mathbf{p} \in \mathcal{W}_D$, the IK problem has at least one solution for any orientation, $\mathcal{W}_D \subseteq \mathcal{W}_R$

⁎ A manipulator is redundant when $\dim(\mathcal{Q}) > \dim(\mathcal{X})$, meaning it has more controllable joint variables than are necessary to specify any pose in the task space. This allows infinitely many solutions to some inverse kinematics problems, enabling optimization for secondary goals (e.g., obstacle avoidance, energy efficiency)

→Lec. 5: Jacobian Matrix $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$

Jacobian Matrix

Consider a smooth differentiable map representing the forward kinematics (FK) of a robot manipulator:

$$f : \mathbb{R}^n \to \mathbb{R}^m, \quad \mathbf{q} \mapsto \mathbf{x} = f(\mathbf{q}),$$

Definition (Jacobian matrix): The Jacobian matrix $\mathbf{J}(\mathbf{q})$ is the matrix of all first-order partial derivatives of f with respect to $\mathbf{q}$:

$$\mathbf{J}(\mathbf{q}) := \frac{\partial f}{\partial \mathbf{q}} \in \mathbb{R}^{m\times n} \qquad \text{where} \qquad \mathbf{J}(\mathbf{q})_{ij} = \frac{\partial x_i}{\partial q_j}$$

In other words, the Jacobian matrix is the derivative of the forward kinematics map f and linearly approximates how small changes in the joint configuration $\mathbf{q}$ affect the end-effector pose $\mathbf{x}$.

⁕ Applying the chain rule to the forward kinematics:

$$\mathbf{x} = f(\mathbf{q}) \Longrightarrow \frac{d\mathbf{x}}{dt} = \frac{df(\mathbf{q})}{dt} = \frac{\partial f}{\partial \mathbf{q}}\frac{d\mathbf{q}}{dt} = \dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

$$\boxed{\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}}$$

where: $\dot{\mathbf{x}} = [{}^0\mathbf{v}_n \quad {}^0\boldsymbol{\omega}_n]^\mathrm{T} \in \mathbb{R}^6$ is the spatial velocity twist of the end-effector combining linear and angular velocity and $\dot{\mathbf{q}}$ are the joint velocities.

Principle of Virtual Work

For an ideal (lossless, no friction) manipulator, the power input by the joints equals the power output at the end-effector for all joint velocities.

$$\boldsymbol{\tau}^\top\dot{\mathbf{q}} = \mathbf{F}^\top\dot{\mathbf{x}} = \mathbf{F}^\top\mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad \forall\dot{\mathbf{q}} \Longrightarrow \boldsymbol{\tau}^\top = \mathbf{F}^\top\mathbf{J}(\mathbf{q}) \Longrightarrow \boldsymbol{\tau} = \mathbf{J}(\mathbf{q})^\top\mathbf{F}$$

where:

$\boldsymbol{\tau} \in \mathbb{R}^n$ generalized forces (torques for revolute, forces for prismatic)

$\mathbf{F} \in \mathbb{R}^6$ wrench applied at the end-effector, combining force and moment

This underlies the principle of virtual work, which equates the virtual work done at the joints and end-effector for any virtual displacement.

⁕ Define ${}^0\mathbf{z}_{j-1}$ as the unit vector along the axis of rotation reltative to the base frame. The jth column of the Jacobian for a revolute and prismatic joints respectively joint is:

$$\text{Revolute} \quad \mathbf{J}_j = \begin{bmatrix} {}^0\mathbf{z}_{j-1} \times ({}^0\mathbf{t}_n - {}^0\mathbf{t}_{j-1}) \\ {}^0\mathbf{z}_{j-1} \end{bmatrix} \qquad \mathbf{J}_j = \begin{bmatrix} {}^0\mathbf{z}_{j-1} \\ \mathbf{0} \end{bmatrix} \quad \text{Prismatic}$$

The full Jacobian matrix for all joints is then: $\mathbf{J}(\mathbf{q}) = [\mathbf{J}_1 \quad \mathbf{J}_2 \quad \cdots \quad \mathbf{J}_n]$

→Lec. 6: Manipulator Dynamic Model

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}\,) + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}$$

❋ In manipulator dynamics, the equations of motion relate joint torques to joint accelerations, velocities, and positions. The mass/inertia matrix $\mathbf{M}(\mathbf{q})$ encodes how the robot's mass distribution resists acceleration. The vector $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}\,)$ collects Coriolis and centrifugal effects, arising naturally because the manipulator's joints form a non-inertial (rotating and moving) reference frame. Finally, the gravity vector $\mathbf{G}(\mathbf{q})$ captures gravitational forces at each joint. These terms together satisfy the Euler-Lagrange equations derived from the manipulator's lagrangian.

$$P = \sum_{i=1}^{n} P_i = -\sum_{i=1}^{n} m_i\ \mathbf{g}^T \mathbf{r}_{i,cm}$$

$$K = \frac{1}{2}\sum_{i=1}^{n} m_i\ \mathrm{tr}\left[\left(\sum_{j=1}^{i} \mathbf{U}_{ij}\,\dot{q}_j\right)\mathbf{r}_{i,cm}\left(\sum_{k=1}^{i} \mathbf{U}_{ik}\,\dot{q}_k\right)^T \mathbf{r}_{i,cm}^T\right] + \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{\omega}_i^T\,\mathbf{I}_{i,cm}\boldsymbol{\omega}_i$$

$$K = \sum_{i=1}^{n} K_i = \sum_{i=1}^{n}\left(\frac{1}{2} m_i \parallel \mathbf{v}_{i,cm} \parallel^2 + \frac{1}{2}\boldsymbol{\omega}_i^T \mathbf{I}_{i,cm}\boldsymbol{\omega}_i\right)$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{i} \mathrm{tr}\left(\mathbf{U}_{ij}\mathbf{J}_{i,cm}\mathbf{U}_{ik}^T\right)\dot{q}_j\dot{q}_k$$

$$\mathbf{v}_{i,cm} = \frac{d}{dt}\left(\mathbf{T}_i\,\mathbf{r}_{i,cm}\right) = \left(\sum_{j=1}^{i} \mathbf{U}_{ij}\,\dot{q}_j\right)\mathbf{r}_{i,cm} \quad \text{where} \quad \mathbf{U}_{ij} := \frac{\partial \mathbf{T}_i}{\partial q_j}$$

$$\mathbf{U}_{ij} = \mathbf{T}_1 \cdots \frac{\partial \mathbf{T}_j}{\partial q_j} \cdots \mathbf{T}_i = \mathbf{A}_1 \cdots \mathbf{Q}_j \mathbf{A}_j \cdots \mathbf{A}_i$$

$$L = K - P = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{i} \mathrm{tr}\left(\mathbf{U}_{ij}\mathbf{J}_{i,cm}\mathbf{U}_{ik}^T\right)\dot{q}_j\dot{q}_k + \sum_{i=1}^{n} m_i\ \mathbf{g}^T\mathbf{T}_i\mathbf{r}_{i,cm}$$

$$\parallel \mathbf{v}_{i,cm} \parallel^2 = \mathrm{tr}\left[\left(\sum_{j=1}^{i} \mathbf{U}_{ij}\,\dot{q}_j\right)\mathbf{r}_{i,cm}\left(\sum_{k=1}^{i} \mathbf{U}_{ik}\,\dot{q}_k\right)^T \mathbf{r}_{i,cm}^T\right]$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \qquad \frac{\partial L}{\partial \dot{q}_i} = \sum_{p=1}^{n}\sum_{j=1}^{p} \mathrm{tr}\left(\mathbf{U}_{pj}\mathbf{J}_{p,cm}\mathbf{U}_{pi}^T\right)\dot{q}_j$$

$$\mathbf{J}_{i,cm} := m_i\mathbf{r}_{i,cm}\mathbf{r}_{i,cm}^T + \mathbf{I}_{i,cm}$$

$$\Rightarrow \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) = \sum_{p=1}^{n}\sum_{j=1}^{p} \mathrm{tr}\left(\mathbf{U}_{pj}\mathbf{J}_{p,cm}\mathbf{U}_{pi}^T\right)\ddot{q}_j + \sum_{p=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{n} \mathrm{tr}\left(\frac{\partial \mathbf{U}_{pj}}{\partial q_k}\mathbf{J}_{p,cm}\mathbf{U}_{pi}^T\right)\dot{q}_j\dot{q}_k$$

$$\mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}, \quad \ddot{\mathbf{q}} = \begin{bmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_n \end{bmatrix}$$

$$= \sum_{p=1}^{n}\sum_{j=1}^{p} \mathrm{tr}\left(\mathbf{U}_{pj}\mathbf{J}_{p,cm}\mathbf{U}_{pi}^T\right)\ddot{q}_j + \sum_{p=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{n} \mathrm{tr}\left(\frac{\partial \mathbf{U}_{pj}}{\partial q_k}\mathbf{J}_{p,cm}\mathbf{U}_{pi}^T\right)\dot{q}_j\dot{q}_k$$

$$\frac{\partial L}{\partial q_i} = \frac{1}{2}\sum_{p=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{p} \mathrm{tr}\left(\frac{\partial \mathbf{U}_{pj}}{\partial q_i}\mathbf{J}_{p,cm}\mathbf{U}_{pk}^T + \mathbf{U}_{pj}\mathbf{J}_{p,cm}\frac{\partial \mathbf{U}_{pk}^T}{\partial q_i}\right)\dot{q}_j\dot{q}_k + \sum_{p=1}^{n} m_p\ \mathbf{g}^T\frac{\partial(\mathbf{T}_p\mathbf{r}_{p,cm})}{\partial q_i}$$

$$D_{ij} := \sum_{p=\max(i,j)}^{n} \mathrm{tr}\left(\mathbf{U}_{pj}\mathbf{J}_{p,cm}\mathbf{U}_{pi}^T\right) \quad D_{ijk} := \sum_{p=\max(i,j,k)}^{n} \mathrm{tr}\left(\frac{\partial \mathbf{U}_{pj}}{\partial q_k}\mathbf{J}_{p,cm}\mathbf{U}_{pi}^T\right) \quad D_i := \sum_{p=i}^{n} m_p\ \mathbf{g}^T\frac{\partial(\mathbf{T}_p\mathbf{r}_{p,cm})}{\partial q_i}$$

$$Q_i = \sum_{j=1}^{n} D_{ij}\,\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} D_{ijk}\,\dot{q}_j\dot{q}_k + D_i + I_{i,act}\ddot{q}_i$$

$$M_{ij} := D_{ij} + I_{i,act}\delta_{ij} \quad \Rightarrow \quad \mathbf{M} \in \mathbb{R}^{n\times n}$$

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) := \begin{bmatrix} \sum_{j=1}^{n}\sum_{k=1}^{n} D_{1jk}\,\dot{q}_j\dot{q}_k \\ \vdots \\ \sum_{j=1}^{n}\sum_{k=1}^{n} D_{njk}\,\dot{q}_j\dot{q}_k \end{bmatrix} \in \mathbb{R}^n, \ \mathbf{G}(\mathbf{q}) := \begin{bmatrix} D_1 \\ \vdots \\ D_n \end{bmatrix}, \ \boldsymbol{\tau} := \begin{bmatrix} Q_1 \\ \vdots \\ Q_n \end{bmatrix}$$

$$\boxed{\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}}$$