Robot Dynamics

Lecturer: Prof. Marco Hutter, Notes: Yunlong Song

Respect is earned through actions, not demanded with words.

Introduction

Robot dynamics is concerned with the relationship between the forces acting on a robot and the accelerations they produce. There are mainly two categories of robot dynamics:

- **Fixed-base**
- **Floating-base**

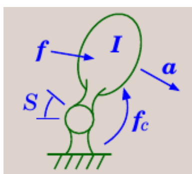


Figure 1: Robot Dynamics. (Source: Roy Featherstone)

Kinematics

Kinematics is the description of the motion of *points*, *bodies*, and *systems* without consideration of the forces that cause the motion.

Position The position of a point can be described in different coordinate systems, such as *Cartesian*, *Cylindrical*, and *Spherical* coordinates.

Cartesian

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Cylindrical

$$\mathbf{r} = \rho\mathbf{e}_\rho + z\mathbf{k}$$

$$\mathbf{r} = \begin{bmatrix} \rho \\ \theta \\ z \end{bmatrix}$$

Spherical

$$\mathbf{r} = r\mathbf{e}_r$$

$$\mathbf{r} = \begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix}$$

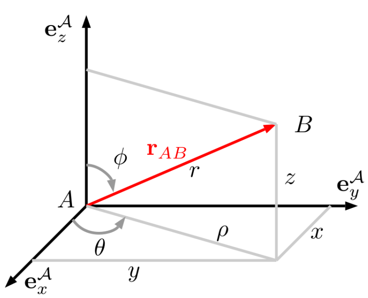


Figure 2: Position

Velocity is the rate of change of position. It is described by the derivative of position with respect to time.

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

Rotation is the orientation of a body in space. It is described by Euler angles, a rotation matrix, or a quaternion.

Euler Angles

$$[\theta, \phi, \psi] = [\text{yaw}, \text{pitch}, \text{roll}]$$

Angle-Axis

$$[\theta, \mathbf{a}] = [\text{angle}, \text{axis}]$$

Rotation Matrix

$$R = R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

Quaternion

$$q = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}$$

Example: Rotation with Quaternions Multiplication Given a vector $\mathbf{v} = [0, 1, 0]$ in frame A. What is the vector in frame B after a rotation of 90° about the z-axis?

$$q = \begin{bmatrix} \cos(\theta/2) \\ 0\sin(\theta/2) \\ 0\sin(\theta/2) \\ \sin(\theta/2) \end{bmatrix} = \begin{bmatrix} 0.7071 \\ 0 \\ 0 \\ 0.7071 \end{bmatrix}$$

$$\mathbf{v}_B = q\mathbf{v}q^*$$

$$\mathbf{v}_B = \begin{bmatrix} 0.7071 \\ 0 \\ 0 \\ 0.7071 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0.7071 \\ 0 \\ 0 \\ -0.7071 \end{bmatrix}$$

Angular Velocity is the rate of change of rotation. It is described by the derivative of rotation with respect to time.

$$\boldsymbol{\omega} = \frac{dR}{dt}R^T = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Relationship between Linear and Angular Velocity

$$\mathbf{v}_B = \mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}_{AB}$$

$$\mathbf{v}_B = \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix} = \begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{z}_A \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} x_{AB} \\ y_{AB} \\ z_{AB} \end{bmatrix}$$

Relationship between Quaternion and Angular Velocity

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} q$$

Transformation

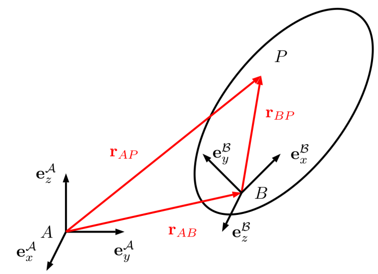


Figure 3: Single body with body frame B and inertial frame A.

Transformation between Frames

$${}_{\mathcal{A}}\mathbf{r}_{AP} = {}_{\mathcal{A}}\mathbf{r}_{AB} + \mathbf{R}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{BP}$$

A homogeneous transformation matrix T is used to describe the relationship between two coordinate frames. It is a 4×4 matrix that combines a rotation matrix R and a translation vector $\mathbf{r}$.

$$\mathbf{T}_{\mathcal{AB}} = \begin{bmatrix} \mathbf{R}_{\mathcal{AB}} & {}_{\mathcal{A}}\mathbf{r}_{AB} \\ 0 & 1 \end{bmatrix}$$

Rigid-body formulation for velocity

$$\mathbf{v}_B = \mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}_{AB}$$

$$\mathbf{v}_B = \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix} = \begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{z}_A \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} x_{AB} \\ y_{AB} \\ z_{AB} \end{bmatrix}$$

where $\mathbf{v}_A$ and $\mathbf{v}_B$ are the linear velocities of points P in frames A and B, respectively. $\boldsymbol{\omega}$ is the angular velocity of frame B with respect to frame A.

Kinematics of Systems of Bodies

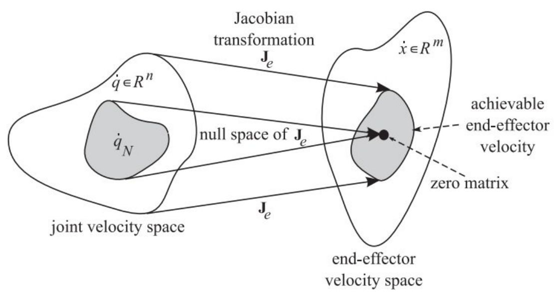


Figure 4: Mapping between Joint Space and Task Space.

Generalized Coordinates are a set of coordinates that completely describe the configuration of a system.

$$\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}^T$$

Degrees of Freedom are the number of independent coordinates needed to describe the configuration of a system.

Task-space Coordinates are the coordinates that describe the position and orientation of the end-effector of a robot. They are derived from the generalized coordinates of the robot.

$$\mathbf{x}_e = \left[\mathbf{r}_e \in \mathbb{R}^3, \phi_e \in SO(3)\right]^T$$

Operational Space Coordinates: The end-effector of a manipulator operates in operational space coordinates—a minimal selection of end-effector configuration parameters.
Example: SCARA Robot with 4 DOF

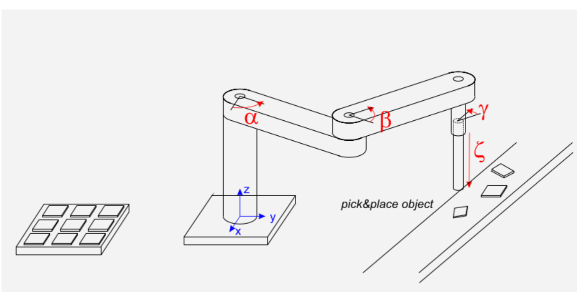


Figure 5: Generalized Coordinates: $\mathbf{q} = [\alpha, \beta, \gamma, \zeta]^T$. Degrees of Freedom: 4. Operational Space Coordinates: *boldsymbol*$\chi_o = [x, y, z, \phi]^T$.

Forward Kinematics is the process of determining the position and orientation of the end-effector of a robot given the joint angles.

$$\boldsymbol{\chi}_e = \boldsymbol{\chi}_e(\mathbf{q})$$

Differential Kinematics is the process of determining the velocity of the end-effector of a robot given the joint velocities.

$$\dot{\boldsymbol{\chi}}_e = J_{eA}(\mathbf{q})\dot{\mathbf{q}}$$

where $J_{eA}(\mathbf{q}) \in \mathbb{R}^{m_e \times n_j}$ is the **Jacobian Matrix** of the robot. It is defined as:

$$J_{eA}(\mathbf{q}) = \begin{bmatrix} \frac{\partial \boldsymbol{\chi}_1}{\partial q_1} & \cdots & \frac{\partial \boldsymbol{\chi}_1}{\partial q_{n_j}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{\chi}_m}{\partial q_1} & \cdots & \frac{\partial \boldsymbol{\chi}_m}{\partial q_{n_j}} \end{bmatrix}$$

Position and Orientation Jacobian

$$J_{eA} = \begin{bmatrix} J_{eA_P} \\ J_{eA_R} \end{bmatrix} = \begin{bmatrix} \frac{\partial \chi_{eP}}{\partial q} \\ \frac{\partial \chi_{eR}}{\partial q} \end{bmatrix}$$

Geometric or Basic Jacobian There must exist a unique Jacobian that relates the generalized velocities $\dot{q}$ to the end-effector velocities (linear $\mathbf{v}_e$ and angular $\boldsymbol{\omega}_e$). This Jacobian is called the *geometric Jacobian* or *basic Jacobian*.

$$\mathbf{w}_e = \begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \underbrace{J_{e0}(\mathbf{q})}_{\text{geometric/basic Jacobian}} \quad \dot{\mathbf{q}}$$

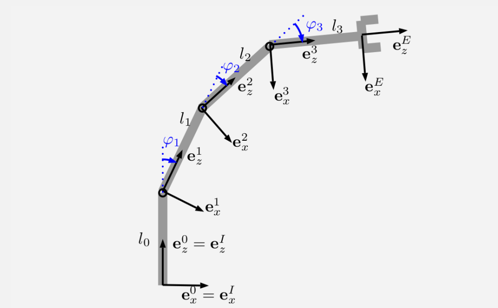
withe $J_{e0}(\mathbf{q}) \in \mathbb{R}^{6 \times n_j}$.

Geometric Jacobian and Analytical Jacobian The following relationship holds between the geometric Jacobian and the analytical Jacobian:

$$J_{e0}(\mathbf{q}) = \mathbf{E}_e(\boldsymbol{\chi}_e) J_{eA}(\mathbf{q})$$

given that $\mathbf{w}_e = \mathbf{E}_e(\boldsymbol{\chi}_e)\dot{\boldsymbol{\chi}}_e$ with $\mathbf{E}_e = \begin{bmatrix} \mathbf{E}_P & 0 \\ 0 & \mathbf{E}_R \end{bmatrix} \in \mathbb{R}^{6 \times m_e}$

Kinematics of a 3-DOF Planar Manipulator



The generalized coordinates are

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{pmatrix}$$

Forward kinematics

$$\chi_e(q) = \begin{pmatrix} \chi_{eP}(q) \\ \chi_{eR}(q) \end{pmatrix}$$

1) The position of the end-effector is

$$\chi_{eP}(q) = \begin{bmatrix} l_1 \sin q_1 + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \\ l_0 + l_1 \cos q_1 + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \end{bmatrix}$$

2) The orientation of the end-effector is

$$\chi_{eR} = q_1 + q_2 + q_3$$

Differential kinematics
The Jacobian matrix is

$$J_{eA} = \begin{bmatrix} \frac{\partial \chi_{eP}}{\partial q} \\ \frac{\partial \chi_{eR}}{\partial q} \end{bmatrix} = \begin{bmatrix} \frac{\partial \chi_{eP}}{\partial q_1} & \frac{\partial \chi_{eP}}{\partial q_2} & \frac{\partial \chi_{eP}}{\partial q_3} \\ \frac{\partial \chi_{eR}}{\partial q_1} & \frac{\partial \chi_{eR}}{\partial q_2} & \frac{\partial \chi_{eR}}{\partial q_3} \end{bmatrix}$$

The position Jacobian is

$$J_{eA_P}(q) = \frac{\partial \chi_{eP}}{\partial q} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \end{bmatrix}$$

with $c_i = \cos q_i$ and $s_i = \sin q_i$. The orientation Jacobian is

$$J_{eA_R} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Kinematics Control Methods

Inverse Differential Kinematics

$$\begin{cases} \dot{\mathbf{w}}_e & = J_{e0}(\mathbf{q})\dot{\mathbf{q}} \quad \text{Forward Differential Kinematics} \\ \dot{\mathbf{q}} & = J_{e0}^{+}(\mathbf{q})\dot{\mathbf{w}}_e \quad \text{Inverse Differential Kinematics} \end{cases}$$

Singularity is a configuration of a robot where the Jacobian matrix $J_{e0}(\mathbf{q})$ has a $\mathrm{rank}(J_{e0}(\mathbf{q})) < m_0$, with m_0 being the number of operational space coordinates (numer of controlable end-effector DOFs).

Moore-Penrose Pseudo-inverse: Damped Solution

$$\dot{\mathbf{q}} = J_{e0}^{+}(\mathbf{q})\dot{\mathbf{w}}_e = (J_{e0}^{T}(\mathbf{q})J_{e0}(\mathbf{q}) + \lambda^2\mathbf{I})^{-1}J_{e0}^{T}(\mathbf{q})\dot{\mathbf{w}}_e$$

Redundancy is a configuration of a robot where the number of degrees of freedom is greater than the number of degrees of freedom needed to perform a task. Redundancy implies that there are multiple solutions to the inverse kinematics problem.

$$\begin{aligned} \dot{\mathbf{q}} &= J_{e0}^{+}(\mathbf{q})\dot{\mathbf{w}}_e + N(\mathbf{q})\dot{\mathbf{q}}_0 \\ &= J_{e0}^{+}(\mathbf{q})\dot{\mathbf{w}}_e + (I - J_{e0}^{+}(\mathbf{q})J_{e0}(\mathbf{q}))\dot{\mathbf{q}}_0 \end{aligned}$$

with $N(\mathbf{q}) = I - J_{e0}^{+}(\mathbf{q})J_{e0}(\mathbf{q})$ being the null space of the Jacobian matrix.

Multi-task Inverse Differential Kinematics The task Jacobian and desired velocity are defined as

$$task_i := \{J_i, \mathbf{w}_i\}$$

Multi-task with Equal Priority: In case all n_t tasks have equal priority, the solution for generalized velocities is

$$\dot{\mathbf{q}} = \underbrace{\begin{bmatrix} J_1 \\ \vdots \\ J_{n_t} \end{bmatrix}^{T}}_{\bar{J}} \underbrace{\left(\begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_{n_t} \end{bmatrix} \right)}_{\bar{\mathbf{w}}}$$

A way to weight some tasks with higher priority is to use the following formulation:

$$\bar{\mathbf{J}}^{+} = (\bar{J}^{T}W\bar{J})^{-1}J^{T}W$$

with $W = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{n_t})$ being a diagonal matrix with the weights λ_i.

Multi-task with Prioritization: Assuming that the tasks are ordered by priority, the solution for generalized velocities is

$$\dot{\mathbf{q}} = \sum_{i=1}^{n_t} \bar{N}_i^{+}\dot{\mathbf{q}}_i \quad \text{with} \quad \dot{\mathbf{q}} = (J_i\bar{N}_i)^{+}\left(\mathbf{w}_i^{*} - J_i\sum_{j=1}^{i-1}\bar{N}_i\dot{\mathbf{q}}_j\right)$$

whereby $\bar{N}_i = I - J_i^{+}(\bar{J}_i)$ is the null space of the task Jacobian.

Inverse Kinematics

$$\begin{aligned} \mathbf{q} &= q(\boldsymbol{\chi}_e^{*}) \\ \dot{\boldsymbol{\chi}}_e &= J_{eA}(\mathbf{q})\dot{\mathbf{q}} \quad \text{analytical solution} \\ \Delta\boldsymbol{\chi}_e &= J_{eA}(\mathbf{q})\Delta\mathbf{q} \quad \text{numerical solution} \end{aligned}$$

Floating Base Kinematics

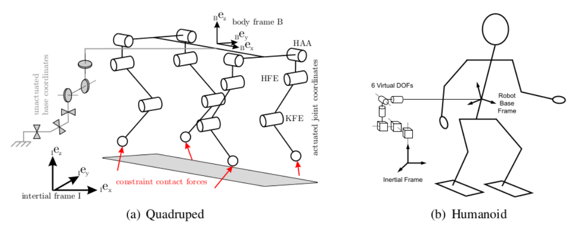


Figure 6: Floating Base Kinematics

Generalized Coordinates: Floating-based kinematics are described by n_b un-actuated base coordinates $\mathbf{q}_b$ and n_j actuated joint coordinates $\mathbf{q}_j$.

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{bmatrix} \quad \text{with} \quad \mathbf{q}_b \in \mathbb{R}^{n_b}, \mathbf{q}_j \in \mathbb{R}^{n_j}$$

Generalized Velocities and Accelerations: The generalized velocities are defined as

$$\mathbf{u} = \begin{bmatrix} {}_I v_B \\ {}_B\omega_{IB} \\ \dot{\varphi}_1 \\ \vdots \\ \dot{\varphi}_{n_j} \end{bmatrix} \in \mathbb{R}^{6+n_j} \quad \dot{\mathbf{u}} = \begin{bmatrix} {}_I a_B \\ {}_B\dot{\phi}_{IB} \\ \ddot{\varphi}_1 \\ \vdots \\ \ddot{\varphi}_{n_j} \end{bmatrix} \in \mathbb{R}^{6+n_j}$$

Linear Mapping $\mathbf{u} = \mathbf{E}_{fb}\dot{\mathbf{q}}$ with

$$\mathbf{E}_{fb} = \begin{bmatrix} \mathbb{I}_{3\times 3} & 0 & 0 \\ 0 & \mathbf{E}_{\chi R} & 0 \\ 0 & 0 & \mathbb{I}_{n_j \times n_j} \end{bmatrix}$$

Dynamics

Dynamics is the study of the relationship between the forces acting on a robot and the accelerations they produce. It is concerned with the motion of the robot as a function of time.

Equations of motion for a rigid body can be written in the general form:

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau + J_c(q)^{T}F_c \tag{1}$$

- $M(q)$ – Mass matrix
- $b(q,\dot{q})$ – Coriolis and centrifugal forces
- $g(q)$ – Gravitational forces
- τ – Generalized forces
- $J_c(q)$ – Contact Jacobian
- F_c – External contact forces

Principle of Virtual Work (D'Alembert's Principle)

$$\delta W = \int_{\mathcal{B}} \delta\mathbf{r} \cdot (\ddot{\mathbf{r}}\mathrm{d}m - \mathrm{d}\mathbf{F}_{ext}) = 0$$

where δW is the virtual work done by the forces acting on the body, $\delta\mathbf{r}$ is the virtual displacement of the body, $\ddot{\mathbf{r}}$ is the acceleration of the body, $\mathrm{d}m$ is the mass of the body, and $\mathrm{d}\mathbf{F}_{ext}$ is the external forces acting on the body.

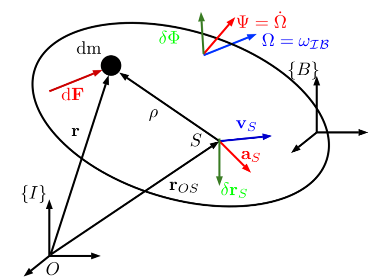


Figure 7: Virtual Displacement of Single Rigid Body

Newton-Euler Method

Newton-Euler Equations are a set of equations that describe the motion of a rigid body. They are derived from Newton's laws of motion and Euler's equations of motion.

Newton's Laws of Motion

$$\sum \mathbf{F} = m\mathbf{a} \text{ (Linear)}$$

It states that the sum of the forces acting on a body is equal to the mass of the body times its acceleration.

Euler's Equations of Motion

$$\mathbf{M} = I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I\boldsymbol{\omega}$$

where $\mathbf{M}$ is the moment acting on the body, I is the inertia of the body, $\boldsymbol{\omega}$ is the angular velocity of the body, and $\dot{\boldsymbol{\omega}}$ is the angular acceleration of the body.

Lagrangian Method

Principle of Least Action The motion of a system is such that the action is minimized. The action is defined as:

$$S = \int_{t_1}^{t_2} L(q, \dot{q}, t)dt$$

where $L(q, \dot{q}, t)$ is the Lagrangian function of the system. Mathematically, the principle of least action can be written as:

$$\delta \int_{t_1}^{t_2} L(q, \dot{q}, t)dt = \delta S = 0$$

Principle of Least Action

The actual path taken by a system between two points in configuration space is such that the integral of the Lagrangian function over time is minimized.

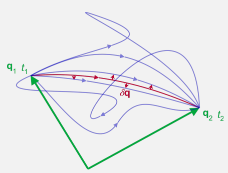


Figure 8: Principle of Least Action

Lagrangian Function

$$L(q, \dot{q}, t) = T - V = T(q, \dot{q}) - V(q)$$
$$T(\text{Kinetic Energy}), \quad V(\text{Potential Energy})$$

Euler-Lagrange Equations

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i$$

This principle examplifies the idea that nature operates in such a way that the is efficient and minimizes the energy expended. The idea of a "least" action is elegant and reflects a deeper, possibly symmetrical structure of the universe.

Example: Lagrangian Dynamics of a 1-DOF Pendulum

- Generalized coordinates: θ
- Kinetic energy: $T = \frac{1}{2}m\ell^2\dot{\theta}^2$
- Potential energy: $V = mgh = m\ell g(1 - \cos\theta)$
- Lagrangian function: $L = T - V$
- Euler-Lagrange equations: $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0$

Solution

- $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = \frac{d}{dt}\left(m\ell^2\dot{\theta}\right) = m\ell^2\ddot{\theta}$
- $\frac{\partial L}{\partial \theta} = \frac{\partial V}{\partial \theta} = m\ell g \sin\theta$
- Euler-Lagrange equation: $m\ell^2\ddot{\theta} + m\ell g \sin\theta = 0$

Hence, the equation of motion of the pendulum is:

$$\ddot{\theta} + \frac{g}{\ell}\sin\theta = 0$$

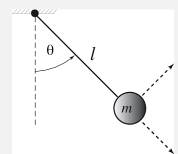


Figure 9: 1-DOF Pendulum

Projected Newton-Euler Method

The projected Newton-Euler equations of motion are derived from the Newton-Euler equations of motion by projecting the forces acting on the robot onto the generalized coordinates of the robot.

Projected Newton-Euler Equations

$$\mathbf{M}_q(q)\ddot{q} + \mathbf{b}_q(q, \dot{q}) + \mathbf{g}_q(q) = \tau_q(q) + \mathbf{J}_c^T(q)\mathbf{F}_c$$

$$\mathbf{M} = \sum_{i=1}^{n_b} \left({}_{\mathcal{A}}\mathbf{J}_{S_i}^T \cdot m \cdot {}_{\mathcal{A}}\mathbf{J}_{S_i} + {}_{\mathcal{B}}\mathbf{J}_{R_i}^T \cdot {}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\mathbf{J}_{R_i}\right)$$
$$\mathbf{b} = \sum_{i=1}^{n_b} \left({}_{\mathcal{A}}\mathbf{J}_{S_i}^T \cdot m \cdot {}_{\mathcal{A}}\dot{\mathbf{J}}_{S_i} \cdot \dot{\mathbf{q}} + {}_{\mathcal{B}}\mathbf{J}_{R_i}^T \cdot \left({}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\dot{\mathbf{J}}_{R_i} \cdot \dot{\mathbf{q}} + {}_{\mathcal{B}}\mathbf{\Omega}_{S_i} \times {}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\mathbf{\Omega}_{S_i}\right)\right)$$
$$\mathbf{g} = \sum_{i=1}^{n_b} \left(-{}_{\mathcal{A}}\mathbf{J}_{S_i}^T {}_{\mathcal{A}}\mathbf{F}_{g,i}\right)$$

Figure 10: Projected Newton-Euler Method

Control

Joint Impedance Control

$$\tau = \mathbf{k}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \hat{g}(\mathbf{q})$$

where τ is the joint torques, $\mathbf{k}_p$ is the proportional gain, $\mathbf{k}_d$ is the derivative gain, $\mathbf{q}_d$ is the desired joint positions, $\mathbf{q}$ is the current joint positions, $\dot{\mathbf{q}}_d$ is the desired joint velocities, $\dot{\mathbf{q}}$ is the current joint velocities, and $\hat{g}(\mathbf{q})$ is the gravity compensation term.

Inverse Dynamics Control

It is based on the inverse dynamics of the robot.

$$\tau = \hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q})$$

where $\hat{\mathbf{M}}(\mathbf{q})$ is the estimated mass matrix, $\hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}})$ is the estimated Coriolis and centrifugal forces, and $\hat{\mathbf{g}}(\mathbf{q})$ is the estimated gravitational forces.

A common approach is to select the desired joint accelerations.

$$\ddot{\mathbf{q}}_d = \mathbf{k}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}})$$

which corresponds to a linear mass-spring-damper system with unitary mass.

Inverse Dynamics Control with Multiple Tasks: Motion in joint space is often hard to describe. Instead, it is easier to describe the motion in task space. The linear and rotational acceleration of the end-effector e is coupled to the generalized accelerations through the *Geometric Jacobians.*

$$\dot{\mathbf{w}}_e = \begin{bmatrix} \ddot{\mathbf{r}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \mathbf{J}_e\ddot{\mathbf{q}} + \dot{\mathbf{J}}_e\dot{\mathbf{q}}$$

End-effector Dynamics and Control

The *End-effector Dynamics* can be described by the following equation:

$$\dot{\mathbf{w}}_e = \mathbf{J}_e\ddot{\mathbf{q}} + \dot{\mathbf{J}}_e\dot{\mathbf{q}} = \mathbf{J}_e\mathbf{M}^{-1}(\tau - \mathbf{b} - \mathbf{g}) + \dot{\mathbf{J}}_e\dot{\mathbf{q}}$$
$$\boldsymbol{\tau} = \mathbf{J}_e^T\mathbf{F}_e$$

yielding the following equation:

$$\mathbf{\Lambda}_e\dot{\mathbf{w}}_e + \boldsymbol{\mu} + \mathbf{p} = \mathbf{F}_e$$

whereby

$$\mathbf{\Lambda}_e = (\mathbf{J}_e\mathbf{M}^{-1}\mathbf{J}_e^T)^{-1}$$
$$\boldsymbol{\mu} = \mathbf{\Lambda}_e\mathbf{J}_e\mathbf{M}^{-1}\mathbf{b} - \mathbf{\Lambda}_e\dot{\mathbf{J}}_e\dot{\mathbf{q}}$$
$$\mathbf{p} = \mathbf{\Lambda}_e\mathbf{J}_e\mathbf{M}^{-1}\mathbf{g}$$

represent the end-effector inertia matrix, the Coriolis and centrifugal forces, and the gravitational forces, respectively, in the task space.

End-effector Control: The end-effector dynamics can be used to control the end-effector.

$$\boldsymbol{\tau} = \hat{\mathbf{J}}_e^T\mathbf{F}_e = \hat{\mathbf{J}}_e^T(\mathbf{\Lambda}_e\dot{\mathbf{w}}_e + \boldsymbol{\mu} + \mathbf{p})$$

The desired end-effector acceleration can be computed as:

$$\dot{\mathbf{w}}_e = \mathbf{k}_p\begin{bmatrix} (\mathbf{r}_{e,d} - \mathbf{r}_e) \\ \Delta\phi_e \end{bmatrix} + \mathbf{k}_d(\dot{\mathbf{w}}_{e,d} - \dot{\mathbf{w}}_e)$$

Please note $\Delta\phi_e$ is the difference between the desired and actual orientation of the end-effector.