Kinematics and Dynamics

Luca Morello

Robot modelling

There are essentially two ways for describing robot movements:

Path

- static collection of points reached by the robot through the motion's execution.

Trajectory

- The point set adds the desired velocity for each point, leading to a path enhanced with temporal data.

Understanding Robot Kinematics

To understand how a robot moves, we study its **mechanics**, **motors**, and **sensors**.

Creating a **mathematical model** helps us control its movement.

Goal: Make the robot perform tasks like picking up objects accurately.

What is Robot Kinematics?

Robot kinematics is the study of **how robots move**.

Focuses only on **position**, **speed**, and **direction** of the robot's parts.

No need to worry about forces (like motor power) – just the movement!

Why is Kinematics Important?

Robots need to know **where** and **how** to move to complete tasks.

Kinematics helps us **calculate exact movements**.

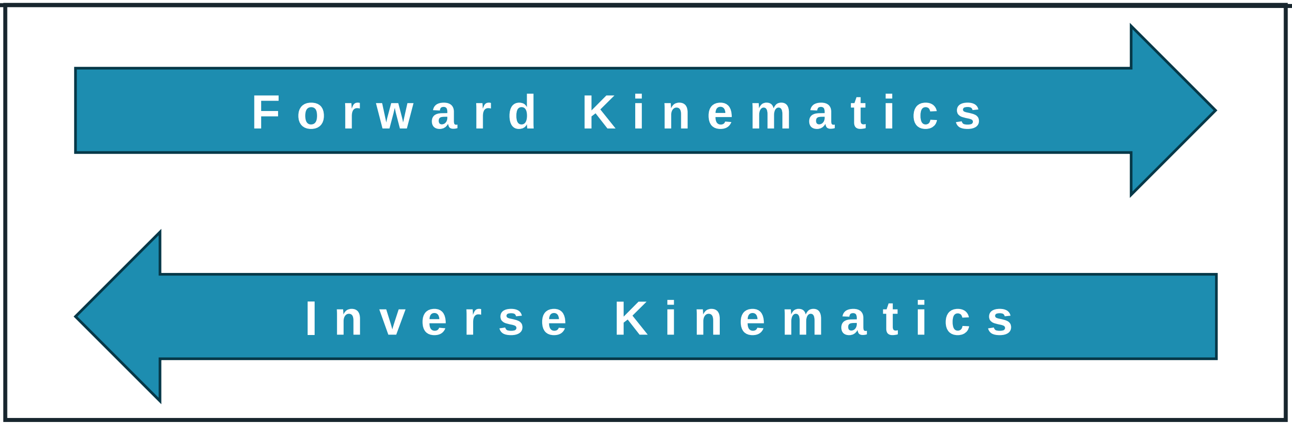
Essential for robots to work **precisely** in various environments (factories, space, etc.).

Key Concepts

Degrees of Freedom (DOF)	Number of **independent movements** a robot can make. Example: A robot arm with **3 DOF** (up/down, left/right, rotation).
Forward Kinematics	Given the **joint angles**, estimates where **the end effector** will move
Inverse Kinematics	Given a **desired position**, calculates the **joint angles** to reach that spot

Joint Space

q_1
q_2
$\vdots$
q_n

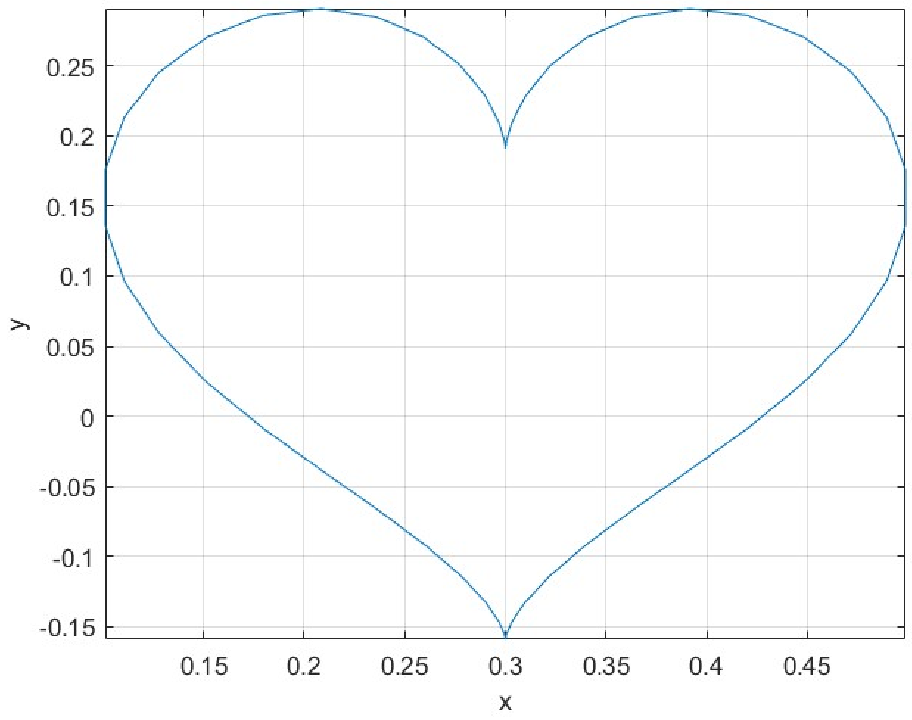


Task Space

x
y
z
$roll$
$pitch$
yaw

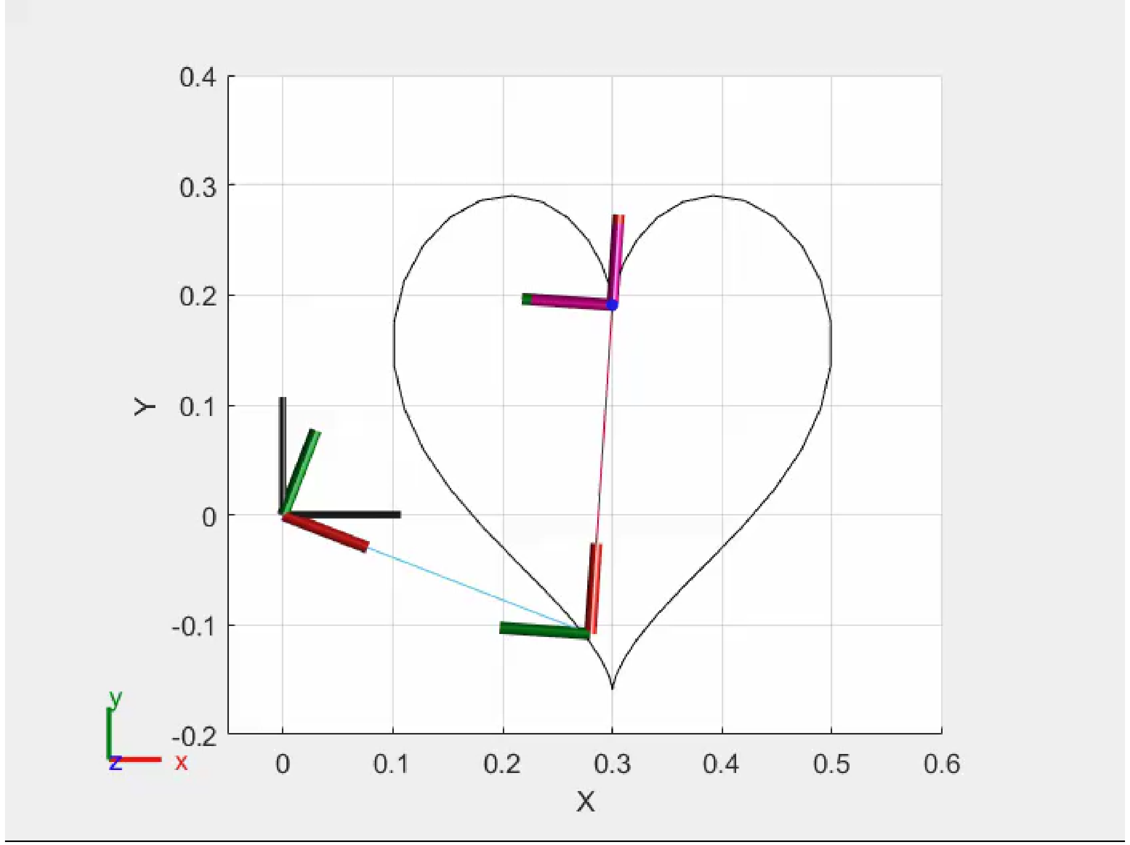
Quick example

We want the robot to follow any assigned path described in Euclidian space.



Quick example

The inverse kinematics of the specified path will extract the robot configurations to follow that path.



Dynamics

Kinematics

- Focuses on the robot's configuration and velocity.
- Describes **joint movements** without regard for the **forces involved** in motion

Limitations in Industrial and Collaborative Robotics:

- In real-world applications, especially with **collaborative robots (cobots)**, forces cannot be ignored.
- Cobots interact with humans and their surroundings, where forces from **collisions, material resistance, and external interactions** are common.

Dynamics

Dynamic Interaction:

- Collaborative robots must be capable of:
  detecting and responding to forces during interactions
- Managing these forces is essential for ensuring
  safety and **efficiency** in environments shared with humans.

Dynamics integrates both **forces** and **torques** with kinematic data

This approach allows for the design of more sophisticated controls that enable the robot to adapt in real-time to changes in applied forces, ensuring precise and safe manipulation.

Dynamics

kinematics should not be neglected

Kinematics is a powerful tool for motion planning, operational evaluation, and trajectory generation

Dynamics enables the robot to "sense" and adapt to force variations, implementing techniques like force compensation or model-based control

Dynamics

- Robot Dynamics
  - Newton’s Law
    - Based on forces and Newton's laws of motion $F = ma$
    - Deals directly with forces acting on objects. Well-suited for simple systems
  - Lagrangian Mechanics
    - Focuses on energy (kinetic, potential) rather than forces
    - Ideal for complex systems with constraints and multiple bodies

Dynamics - **Lagrangian Mechanics**

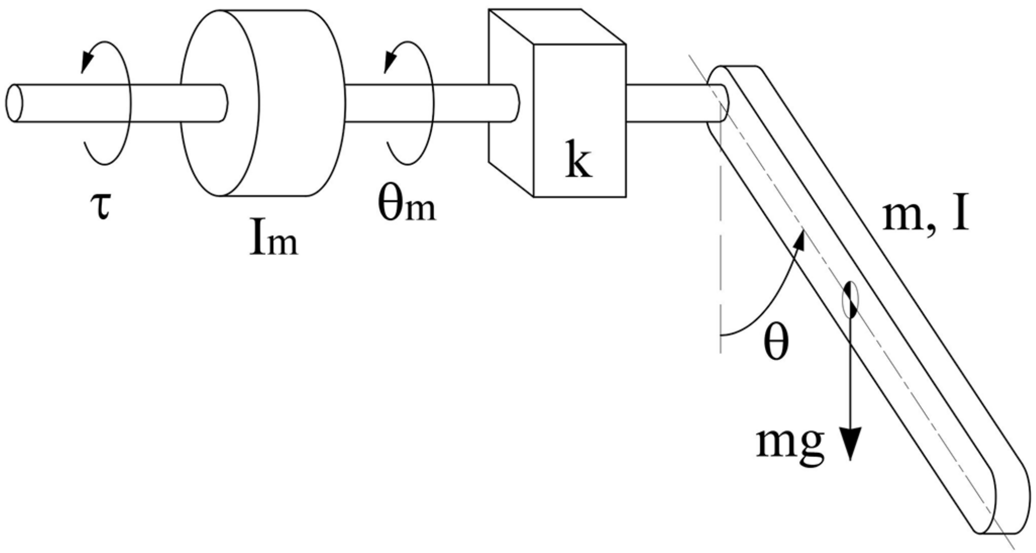
1. Focuses on the **energy** of the system rather than forces directly.
2. Suitable for systems with **complex constraints** and **non-linear behavior** (Very common in robotics)

Key Concepts	
	The Lagrangian is defined as the difference between **kinetic energy (T)** and **potential energy (U)**: $$L = T - U$$
	This formulation allows for the derivation of the **equations of motion** that govern the robot's dynamics.

Dynamics - **Lagrangian Mechanics**

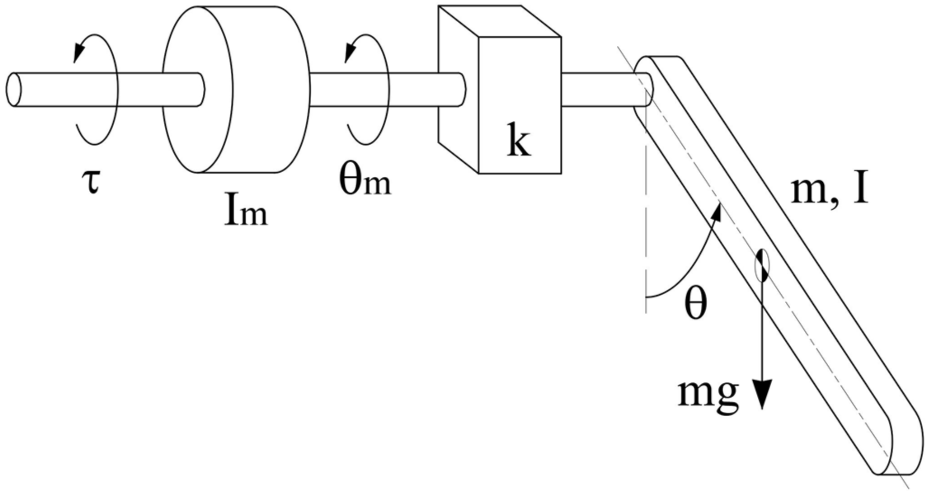
Pendulum example

$\theta = {}^{\theta_m}/_{k}$	Output velocity (motor + reducer)
I_m	Moment of inertia of the rotor
k	Reduction ratio of the reducer
m, l, I	mass, length inertia respectively of the output shaft



Dynamics - **Lagrangian Mechanics**

The Lagrangian is defined as the difference between **kinetic energy (T)** and **potential energy (U)**:



potential Energy

- $U = \frac{l}{2}\, mg(1 - \cos\theta) = \frac{l}{2}\, mg\left(1 - \cos\frac{\theta_m}{k}\right)$

Kinetic Energy

- $T = \frac{1}{2}\, I_m \dot{\theta}_m^2 + \frac{l}{2}\, I\left(\frac{\dot{\theta}_m}{k}\right)^2$

Dynamics - **Lagrangian Mechanics**

Lagrangian: $L = T - U$

$$L = \frac{1}{2} I_m \dot{\theta}_m^2 + \frac{1}{2} I \left(\frac{\dot{\theta}_m}{k}\right)^2 - \frac{l}{2} mg \left(1 - \cos\frac{\theta_m}{k}\right)$$

To obtain the equations of motion for each degree of freedom, we **use Lagrange's equation**

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

Where ξ is the generalized force vector

Dynamics - **Lagrangian Mechanics**

Since we have a simple pendulum with one degree of freedom ($q = \theta_m$)
we will have only a single equation of motion.

We substitute this $\longrightarrow$

$$L = \frac{1}{2} I_m \dot{\theta}_m^2 + \frac{1}{2} I \left(\frac{\dot{\theta}_m}{k}\right)^2 - \frac{l}{2} mg \left(1 - \cos\frac{\theta_m}{k}\right)$$

Into the equation

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

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \xi \quad \rightarrow \quad \left(\frac{\partial L}{\partial \dot{q}}\right) = I_m \dot{\theta}_m + \frac{I}{k^2}\dot{\theta}_m$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \xi \quad \rightarrow \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) = I_m \ddot{\theta}_m + \frac{I}{k^2}\ddot{\theta}_m$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \xi \quad \rightarrow \quad -\left(\frac{\partial L}{\partial q}\right) = \frac{mgl}{2k} sin\frac{\theta_m}{k}$$

Dynamics - Lagrangian Mechanics

Let's explain what the generalized force ξ is.

Generalized force operating on a generalized coordinate q, ξ represents the force or torque (depending on the type of joint) that can perform work on that coordinate.

In our example, the force at the joint is the motor's torque, which must be decreased by frictional loss.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \xi \quad \rightarrow \quad \xi = \tau - \tau_{friction}$$

Dynamics - **Lagrangian Mechanics**

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

$$I_m \ddot{\theta}_m + \frac{I}{k^2}\ddot{\theta}_m + \frac{mgl}{2k} sin \frac{\theta_m}{k} = \tau - \tau_{friction}$$

The procedure we've seen applies to a system with one degree of freedom (DOF).

For more complex systems, the process is the same, but we’ll have n equations for n DOF

Because we describe the dynamics acting on each joint separately.

Dynamics - **Lagrangian Mechanics**

Given a robot with N DoFs is convenient to write this system of equations in matrix form

$$M(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\dot{\theta} + G(\theta) = \xi$$

$M(\theta)$	Inertia matrix	Inertial properties of the robot
$C\left(\theta, \dot{\theta}\right)$	Coriolis matrix	Centrifugal forces
$G(\theta)$	Gravitational matrix	Gravitational forces
ξ	Vector of generalized forces at joints	

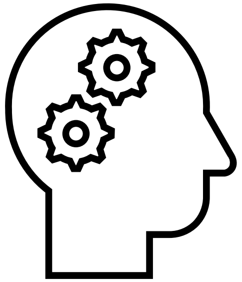
Dynamics → Static

Robot Dynamics: Provides insight into the robot's behavior, at **torques at the joints level**

Collaborative Robots (Cobots): Can adapt to **external forces** and adjust torques to ensure the **end-effector force** follows a reference.

Dynamics → Static

How can we then control the force at the end effector?



Dynamics → Static

An additional step needed to convert **external forces** into useful control information.

The **static equilibrium** condition is described by the formula:

$$\tau = J^T F_{external}$$

This equation allows the robot to translate external forces at the end-effector into joint torques for control.

Static

What Does This Equation Mean?

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{F}$$

Expresses the relationship between **external forces** acting on the end effector and **joint torques**.

$$\boldsymbol{\tau}$$

Vector of **torques** that must be **applied** at the robot joints **to balance** the **forces** at the **end effector**.

$$\boldsymbol{J}^T$$

The **Jacobian matrix transpose** that converts external forces at the end effector into torques at the joints.

- The Jacobian transpose maps the end effector's force space to the joint torque space.

$$\boldsymbol{F}$$

External forces applied to the robot's **end effector**, such as those generated by a gripper or tool.

Static – The Principle of Virtual Forces

According to this theory, the work done by virtual forces must match the work produced by virtual torques at the joints.

The idea is that the power generated by a **virtual force** applied to the end effector is equal to the power that force generates at the robot's joints.

The power produced by an external force F applied to the end effector is:

$$P_{force} = F^T \cdot \dot{x}$$

End effector linear velocity

The power produced by torques at the joints is:

$$P_{torque} = \tau^T \cdot \dot{\theta}$$

Joints angular velocity

Static – The Principle of Virtual Forces

We know from kinematics that:

$$\dot{x} = J\dot{\theta}$$

Equating the two powers and substituting $\dot{x}$ into the first expression, we get:

$$F^T \cdot J\dot{\theta} = \tau^T \cdot \ \dot{\theta}$$

We could simplify the equation in order to find the relationship between the forces on the end effector and the torques at the joints.

This equation shows how external forces at the end effector are transmitted to the joints via the Jacobian matrix providing insight into the system's behavior under static conditions.

$$\tau = J^T F$$

Static – Summary

Why is This Equation Important?

Control: Learning about this equation allows us to control the robot by specifying the desired forces at the end effector and figuring out the joint torques required to achieve them.

Applications:

- **Gripping**: If the robot is picking up an object, we must determine the forces required at the gripper and calculate the joint torques to apply.
- **Manipulation**: To precisely manipulate objects, we use this equation to ensure that the robot applies the appropriate forces at the end effector and then controls the joints accordingly.

Conclusion