Autonomous Mobile Robots

Lecture 3: Control & Perception Fundamentals

Dr. Aliasghar Arab

NYU Tandon School of Engineering

Fall 2025

Today's Topics:

- Recap: Differential Drive Kinematics
- Robot Dynamics via Euler-Lagrange Method
- Example: One-Wheeled Balancing Robot
- Perception Fundamentals
- Coupled Vehicle-Manipulator Systems
- Coordinate Transformations & Swerve Drive

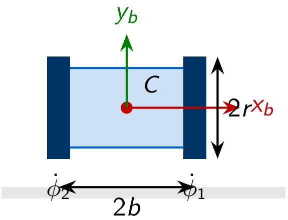
1. Introduction & Physical Constraints
2. Vehicle Kinematics & Equations of Motion
3. **Control & Perception Fundamentals**
3. Nonlinear Control Methods
4. Advanced Model Predictive Control (NMPC) + Constraints
5. Motion Planning Algorithms (RRT*, RRT, DWA)
6. Learning-Based Planning & Control
7. Industry Standards, Architectures, & Safety

Part I

Differential Drive Kinematics Recap

Physical Parameters:

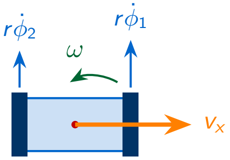
- Two wheels with radius r
- Distance between wheels: $2b$
- Wheel angular velocities:
  - $\dot{\phi}_1$ (right wheel)
  - $\dot{\phi}_2$ (left wheel)



From wheel speeds to body velocities:

Linear velocity (forward):

$$\boxed{v_x = \frac{r}{2}\left(\dot{\phi}_1 + \dot{\phi}_2\right)}$$



Angular velocity (yaw rate):

$$\boxed{\omega = \frac{r}{2b}\left(\dot{\phi}_1 - \dot{\phi}_2\right)}$$

Nonholonomic Constraint

Lateral velocity is always zero: $v_y = 0$ (robot cannot move sideways)

Robot pose in global frame:

$$\boldsymbol{X} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

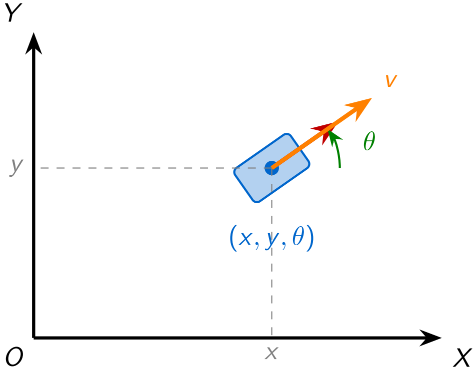
Motion equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Compact form:

$$\boxed{\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})}$$

where $\boldsymbol{u} = [v, \omega]^T$ is the control input.



Y
v
θ
y
(x, y, θ)
O
x
X

Part II

Robot Dynamics

General dynamic model:

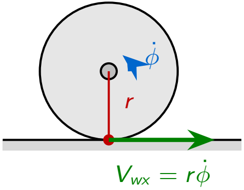
$$\boxed{\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \mathbf{B}_x(\boldsymbol{q})\boldsymbol{F}_x + \mathbf{B}_y\boldsymbol{F}_y}$$

- $\mathbf{M}(\boldsymbol{q})$: Mass/inertia matrix
- $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$: Coriolis/centrifugal
- $\boldsymbol{G}(\boldsymbol{q})$: Gravity terms
- $\mathbf{B}_x, \mathbf{B}_y$: Input mappings
- $\boldsymbol{F}_x, \boldsymbol{F}_y$: Wheel forces
- $\boldsymbol{q}$: Generalized coordinates

No Slipping (Longitudinal):

$$r\dot{\phi} = V_{wx}$$

Wheel rolls without slipping in direction of motion.



No Sliding (Lateral):

$$V_{wy} = 0 \quad \Rightarrow \quad F_y = 0$$

No sideways velocity at wheel contact.

Part III

One-Wheeled Balancing Robot

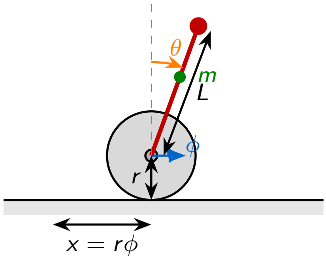
Euler-Lagrange Derivation

System Parameters:

- Wheel radius: r
- Body length: L
- Total mass: m

Generalized Coordinates:

- ϕ: Wheel angle
- θ: Body tilt from vertical



Wheel position:

$$x = r\phi$$

Center of mass position:

Horizontal:

$$\boxed{x_c = x + L\sin\theta = r\phi + L\sin\theta}$$

Vertical:

$$\boxed{z_c = L\cos\theta + r}$$

Velocities:

$$\dot{x}_c = r\dot{\phi} + L\dot{\theta}\cos\theta$$

$$\dot{z}_c = -L\dot{\theta}\sin\theta$$

Total kinetic energy:

$$K = \underbrace{\frac{1}{2} m \left(\dot{x}_c^2 + \dot{z}_c^2\right)}_{\text{translational}} + \underbrace{\frac{1}{2} I_w \dot{\phi}^2}_{\text{wheel rotation}} + \underbrace{\frac{1}{2} I_b \dot{\theta}^2}_{\text{body rotation}}$$

Substituting velocity expressions:

$$\dot{x}_c^2 = \left(r\dot{\phi} + L\dot{\theta}\cos\theta\right)^2$$

$$\dot{z}_c^2 = L^2\dot{\theta}^2 \sin^2\theta$$

$$\dot{x}_c^2 + \dot{z}_c^2 = r^2\dot{\phi}^2 + 2rL\dot{\phi}\dot{\theta}\cos\theta + L^2\dot{\theta}^2$$

Potential energy:

$$V = mgz_c = mg(r + L\cos\theta)$$

Taking reference at $\theta = 0$ (upright):

$$\boxed{V = mgL(\cos\theta - 1)}$$

Lagrangian:

$$\boxed{\mathcal{L} = K - V}$$

Euler-Lagrange Equation

For each generalized coordinate q_i:

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

Step 1: Compute $\frac{\partial \mathcal{L}}{\partial \dot{\phi}}$

$$\frac{\partial \mathcal{L}}{\partial \dot{\phi}} = mr^2\dot{\phi} + mrL\dot{\theta}\cos\theta + I_w\dot{\phi}$$

Step 2: Compute time derivative

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) = (mr^2 + I_w)\ddot{\phi} + mrL\ddot{\theta}\cos\theta - mrL\dot{\theta}^2\sin\theta$$

Step 3: $\frac{\partial \mathcal{L}}{\partial \phi} = 0$ (no explicit ϕ dependence)

Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) - \frac{\partial \mathcal{L}}{\partial \phi} = \tau$$

First Equation of Motion (EOM-1):

$$\boxed{(mr^2 + I_w)\ddot{\phi} + mrL\cos\theta\,\ddot{\theta} - mrL\sin\theta\,\dot{\theta}^2 = \tau}$$

where τ is the motor torque applied to the wheel.

Step 1: Compute $\frac{\partial \mathcal{L}}{\partial \dot{\theta}}$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = mL^2\dot{\theta} + mrL\dot{\phi}\cos\theta + I_b\dot{\theta}$$

Step 2: Time derivative

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) = (mL^2 + I_b)\ddot{\theta} + mrL\ddot{\phi}\cos\theta - mrL\dot{\phi}\dot{\theta}\sin\theta$$

Step 3: Compute $\frac{\partial \mathcal{L}}{\partial \theta}$

$$\frac{\partial \mathcal{L}}{\partial \theta} = -mrL\dot{\phi}\dot{\theta}\sin\theta + mgL\sin\theta$$

Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = 0$$

(No external torque on θ — only gravity acts)

Second Equation of Motion (EOM-2):

$$\boxed{(mL^2 + I_b)\ddot{\theta} + mrL\cos\theta\,\ddot{\phi} - mgL\sin\theta = 0}$$

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

Mass Matrix:

$$\mathbf{M}(\theta) = \begin{bmatrix} mr^2 + I_w & mrL\cos\theta \\ mrL\cos\theta & mL^2 + I_b \end{bmatrix}$$

Coriolis Matrix:

$$\mathbf{C}(\theta, \dot{\theta}) = \begin{bmatrix} 0 & -mrL\sin\theta\,\dot{\theta} \\ 0 & 0 \end{bmatrix}$$

Gravity Vector:

$$\boldsymbol{G}(\theta) = \begin{bmatrix} 0 \\ -mgL\sin\theta \end{bmatrix}$$

Input Vector:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$

Complete Dynamic Equation:

$$\boxed{\begin{bmatrix} mr^2 + I_w & mrL\cos\theta \\ mrL\cos\theta & mL^2 + I_b \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} -mrL\sin\theta\,\dot{\theta}^2 \\ -mgL\sin\theta \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}}$$

Key Properties

- $\mathbf{M}(\theta)$ is symmetric and positive definite
- Coupling through off-diagonal term $mrL\cos\theta$
- Equilibrium at $\theta = 0$ is **unstable**

Control Objective

Design torque τ to:

- Stabilize body upright: $\theta \to 0$
- Control position: $x = r\phi \to x_{\text{desired}}$

Part IV

Perception Fundamentals

The Problem:

- Dynamics give us motion equations
- But we need to know where the robot **actually is**
- Sensors are noisy and imperfect

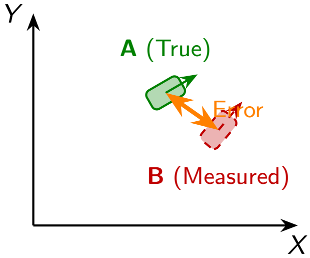
Definition

Perception is the process of estimating the robot's pose (position and orientation) relative to the world using sensor measurements.

Error Components:

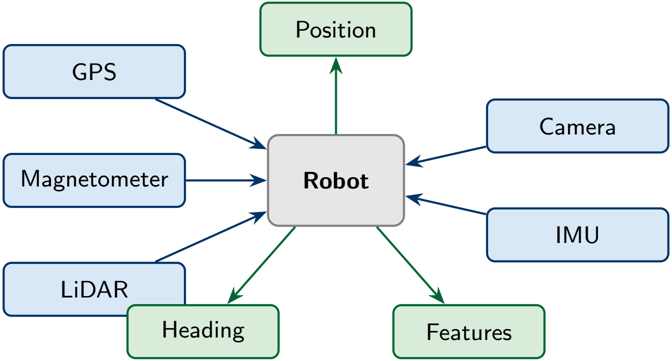
$$\boldsymbol{e} = \boldsymbol{X}_{\text{true}} - \boldsymbol{X}_{\text{measured}}$$

$$\boldsymbol{e} = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}$$



Error Sources:

- Sensor noise
- Sensor bias
- Drift over time



Sensor Fusion: Combining multiple sensors for robustness

GPS:

- Global position (lat, lon)
- Accuracy: 1–10 m typical

Magnetometer:

- Absolute heading
- Subject to interference

IMU:

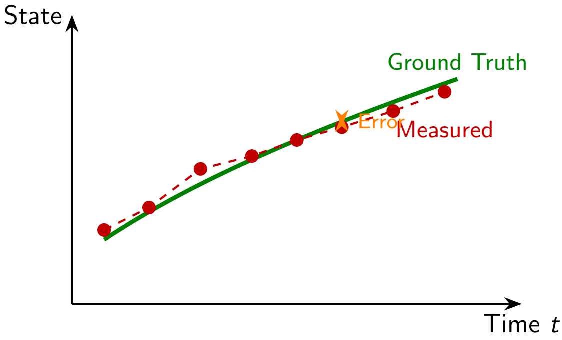
- Accelerations, angular rates
- High rate, but drifts

LiDAR:

- Point cloud data
- Scan matching for localization

Camera:

- Feature-based localization
- Visual SLAM



Key Insight

There's always a gap between what's real and what sensors report.
Perception = estimating and correcting this gap.

Global Frame $\{G\}$:

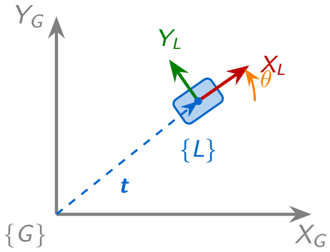
- Fixed world reference
- Used for mapping/navigation

Local Frame $\{L\}$:

- Attached to robot
- Moves with robot

Transformation:

$$\boldsymbol{p}^G = \mathbf{R}(\theta)\boldsymbol{p}^L + \boldsymbol{t}$$



Original (ideal) model:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$$

With uncertainties:

$$\boxed{\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{w}(t)}$$

where $\boldsymbol{w}(t)$ models:

- Process noise (unmodeled dynamics)
- Sensor drift
- Environmental disturbances

Measurement model:

$$\boldsymbol{y} = h(\boldsymbol{x}) + \boldsymbol{v}(t)$$

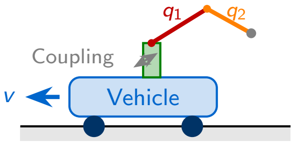
where $\boldsymbol{v}(t)$ is measurement noise.

Part V

Coupled Vehicle-Manipulator Systems

System Configuration:

- Mobile base (vehicle)
- Mounted manipulator (SCARA arm)
- **Dynamics are coupled!**



Key Insight

When a manipulator is mounted on a vehicle, their dynamics are **not independent**.

Manipulator dynamics:

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

- $\mathbf{M}_a$: Inertia of arm
- $\mathbf{C}_a$: Coriolis/centrifugal
- $\boldsymbol{G}_a$: Gravity of arm
- $\boldsymbol{q}_a$: Joint angles
- $\boldsymbol{\tau}_a$: Joint torques

Mobile base dynamics:

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

- $\mathbf{M}_v$: Inertia of vehicle
- $\mathbf{C}_v$: Coriolis/centrifugal
- $\boldsymbol{G}_v$: Gravitational effects
- $\boldsymbol{q}_v$: Vehicle config (x, y, θ)
- $\boldsymbol{\tau}_v$: Driving forces/torques

Full state vector:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_v \\ \boldsymbol{q}_a \end{bmatrix}$$

Unified dynamics:

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

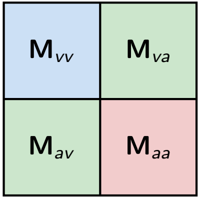
where $\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_v \\ \boldsymbol{\tau}_a \end{bmatrix}$

Important

Cannot write as two independent systems — coupling terms arise from interaction.

Block matrix form:

$$\mathbf{M}(\boldsymbol{q}) = \begin{bmatrix} \mathbf{M}_{vv} & \mathbf{M}_{va} \\ \mathbf{M}_{av} & \mathbf{M}_{aa} \end{bmatrix}$$



- $\mathbf{M}_{vv}$: Vehicle inertia
- $\mathbf{M}_{aa}$: Arm inertia
- $\mathbf{M}_{va}, \mathbf{M}_{av}$: **Coupling**

$\mathbf{M}_{av} = \mathbf{M}_{va}^T$ (symmetry)

Physical interpretation:

- $\mathbf{M}_{va}$: How manipulator motion affects vehicle
- $\mathbf{M}_{av}$: How vehicle motion affects manipulator

Example

When the arm swings, it creates reaction forces on the vehicle.
When the vehicle accelerates, the arm experiences inertial forces.

Controller must account for these coupling effects!

For mecanum wheel robots:

- 3 or 4 mecanum wheels
- Extra degrees of freedom (holonomic)
- Vehicle inertia matrix increases: $3 \times 3 \rightarrow 4 \times 4$

Same block structure applies:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{vv}^{n \times n} & \mathbf{M}_{va} \\ \mathbf{M}_{av} & \mathbf{M}_{aa}^{m \times m} \end{bmatrix}$$

Only the dimensions change — the framework generalizes.

Summary

- Vehicle and manipulator dynamics are **not independent**
- Must model **coupling** between them
- Block matrix structure reveals interactions

Framework generalizes to:

- Any manipulator-on-mobile-base system
- Robotic arms on AGVs
- Dual-arm mobile manipulators

Part VI

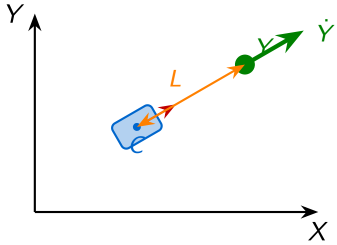
Coordinate Transformations

Define point Y ahead of robot:

$$\boldsymbol{Y} = \begin{bmatrix} x + L\cos\theta \\ y + L\sin\theta \end{bmatrix}$$

Velocity of point Y:

$$\dot{\boldsymbol{Y}} = \begin{bmatrix} v\cos\theta - L\omega\sin\theta \\ v\sin\theta + L\omega\cos\theta \end{bmatrix}$$



Matrix form:

$$\dot{\mathbf{Y}} = \mathbf{J}(\theta) \begin{bmatrix} v \\ \omega \end{bmatrix}$$

where the Jacobian is:

$$\mathbf{J}(\theta) = \begin{bmatrix} \cos\theta & -L\sin\theta \\ \sin\theta & L\cos\theta \end{bmatrix}$$

Key property:

$$\det(\mathbf{J}) = L\cos^2\theta + L\sin^2\theta = L \neq 0$$

The Jacobian is always invertible for $L > 0$!

Inverse relation:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}^{-1}(\theta)\, \dot{\mathbf{Y}}$$

$$\mathbf{J}^{-1}(\theta) = \frac{1}{L} \begin{bmatrix} L\cos\theta & L\sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Control Advantage

Control the look-ahead point Y instead of robot center.
Makes trajectory tracking simpler (feedback linearization).

Define transformed (barred) matrices:

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

$$\bar{\mathbf{C}} = (\bar{\mathbf{J}}^{-1})^T \left(\mathbf{C} - \mathbf{M} \bar{\mathbf{J}}^{-1} \dot{\bar{\mathbf{J}}} \right) \bar{\mathbf{J}}^{-1}$$

Final dynamics at point Y:

$$\boxed{\bar{\mathbf{M}} \ddot{\boldsymbol{Y}} + \bar{\mathbf{C}} \dot{\boldsymbol{Y}} + \bar{\boldsymbol{G}} = \bar{\mathbf{B}} \boldsymbol{F}}$$

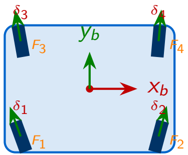
This enables direct control of the look-ahead point position.

Part VII

4-Independent Wheel Drive (Swerve)

Configuration:

- 4 independently controlled wheels
- Each wheel has 2 actuations:
  - Steering angle δ_i
  - Driving force F_i
- Total: 8 control inputs

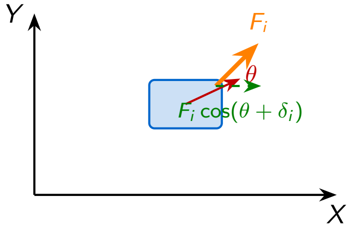


Advantages:

- Holonomic motion
- High maneuverability

Force balance in x:

$$m\ddot{x} = \sum_{i=1}^{4} F_i \cos(\theta + \delta_i)$$



Force balance in y:

$$m\ddot{y} = \sum_{i=1}^{4} F_i \sin(\theta + \delta_i)$$

Each wheel contributes:

- $F_i \cos(\theta + \delta_i)$ in global X
- $F_i \sin(\theta + \delta_i)$ in global Y

Robot motion = combined effect of all 4 wheels.

Moment balance about center of mass:

$$I_{zz}\ddot{\theta} = \sum_{i=1}^{4} \boldsymbol{r}_i \times \boldsymbol{F}_i$$

Expanded:

$$I_{zz}\ddot{\theta} = \sum_{i=1}^{4} [x_i F_i \sin(\theta + \delta_i) - y_i F_i \cos(\theta + \delta_i)]$$

where (x_i, y_i) is wheel i position relative to center.

Control Allocation

Given desired $(\ddot{x}, \ddot{y}, \ddot{\theta})$, solve for F_i and δ_i.
Over-actuated system with redundancy.

State: $\boldsymbol{q} = [x, y, \theta]^T$

Dynamics:

$$\mathbf{M}\ddot{\boldsymbol{q}} = \mathbf{B}(\boldsymbol{q}, \delta)\boldsymbol{F}$$

where:

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \cos(\theta + \delta_1) & \cos(\theta + \delta_2) & \cos(\theta + \delta_3) & \cos(\theta + \delta_4) \\ \sin(\theta + \delta_1) & \sin(\theta + \delta_2) & \sin(\theta + \delta_3) & \sin(\theta + \delta_4) \\ r_1^\perp & r_2^\perp & r_3^\perp & r_4^\perp \end{bmatrix}$$

Numerical integration (Euler method):

$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + \dot{\boldsymbol{x}}(t) \cdot \Delta t$$

Simulation parameters:

- Duration: $T = 10$ s
- Step size: $\Delta t = 0.01$ s
- Tools: MATLAB, Simulink, or Python

Goal:

- Verify simulated path matches expectations
- Test: straight line, circular arc, figure-8

Dynamics:

- Euler-Lagrange formulation
- Balancing robot example
- Matrix form: $\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{G} = \boldsymbol{\tau}$

Perception:

- Sensor types
- Ground truth vs. measurement
- Uncertainty modeling

Coupled Systems:

- Vehicle-manipulator coupling
- Block matrix structure

Transformations:

- Look-ahead point control
- Jacobian transformations
- Swerve drive dynamics

End of Lecture 3

Control & Perception Fundamentals

Questions?