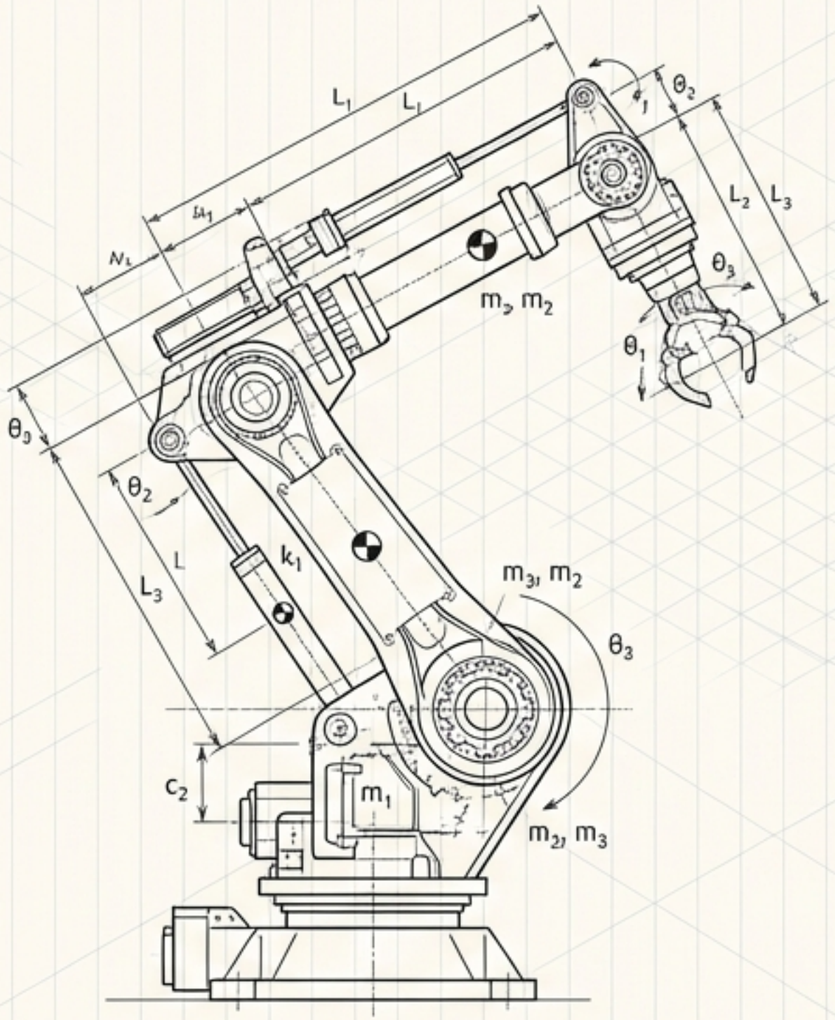
From Physics to Matrices: The Control Engineer's Blueprint

Translating Mechanical Dynamics into Mathematical Models



$$[M] = \begin{bmatrix} (m_1+m_2+m_3)L_1^2 + J_1 + J_3 & (m_2+m_3)L_1L_2\cos(\theta_2-\theta_1) & m_3L_1L_3\cos(\theta_3-\theta_1) \\ (m_2+m_3)L_1L_2\cos(\theta_2-\theta_1) & (m_2+m_3)L_2^2 + J_2 + J_3 & m_3L_2L_3\cos(\theta_3-\theta_2) \\ m_3L_1L_3\cos(\theta_3-\theta_1) & m_3L_2L_3\cos(\theta_3-\theta_2) & m_3L_3^2 + J_3 \end{bmatrix}$$

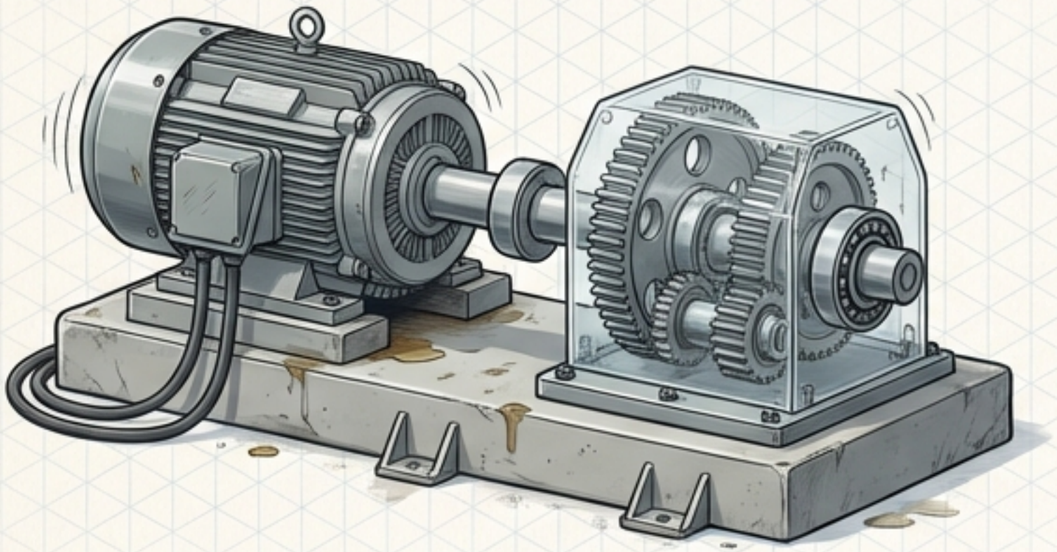
$$+ [C] = \begin{bmatrix} c_1 + c_2 + c_3 & -c_2 & -c_3 \\ -c_2 & c_2 + c_3 & -c_3 \\ -c_3 & -c_3 & c_3 \end{bmatrix} [\dot{\theta}]$$

$$+ [K] = \begin{bmatrix} k_1 + k_2 + k_3 & -k_2 & -k_3 \\ -k_2 & k_2 + k_3 & -k_3 \\ -k_3 & -k_3 & k_3 \end{bmatrix} [\theta] = [F]$$

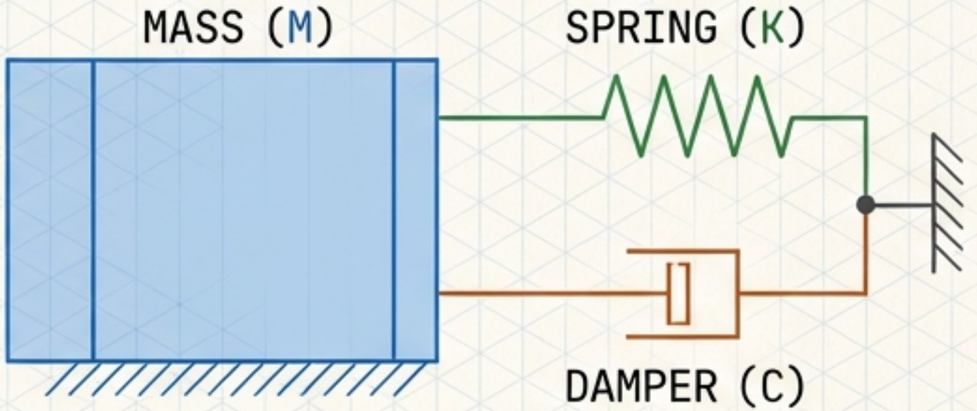
NotebookLM

The Map is Not the Territory

THE TERRITORY



THE MAP



≠

PHYSICAL REALITY

Physical systems are infinitely complex. Analyzing their raw, unpredictable dynamics in the real world is risky, expensive, and subject to environmental noise.

MATHEMATICAL MODEL

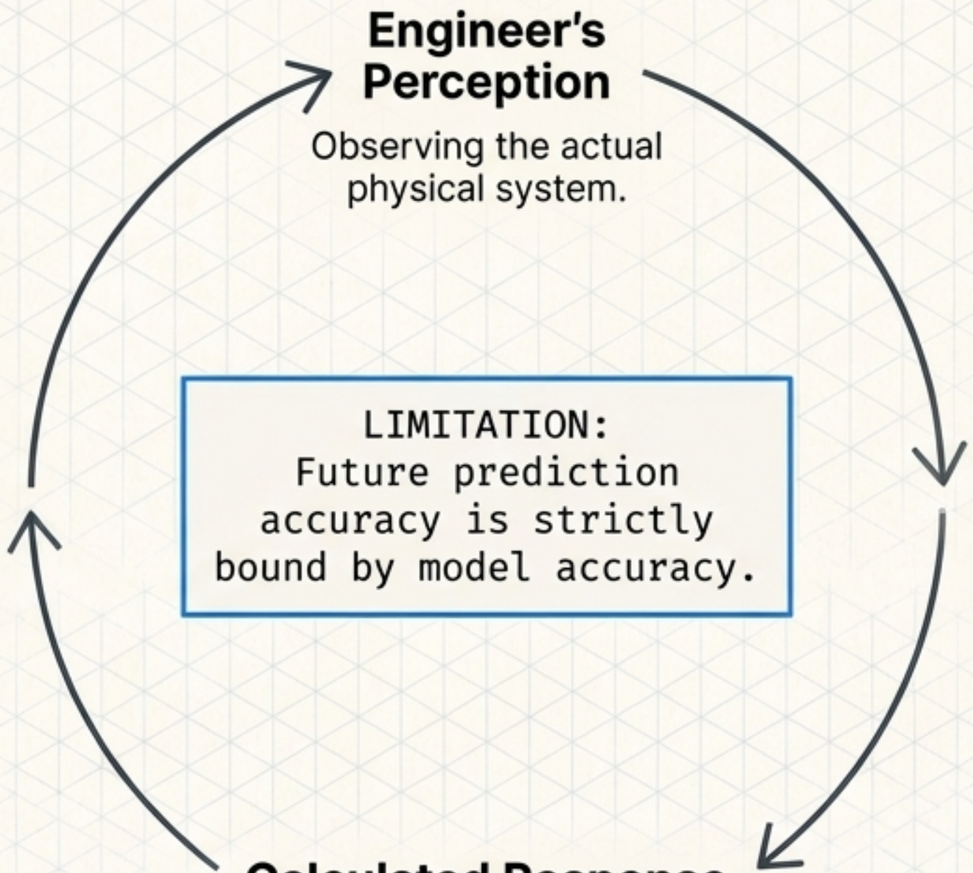
A mathematical model is a deliberate simplification. Like a map, it strips away physical noise to capture only the essential dynamic relationships required for navigation, analysis, and control.

Mathematical Modelling: Deriving equations from fundamental physical laws (Newton, Kirchhoff).

System Identification: Constructing models empirically from measured input-output experimental data.

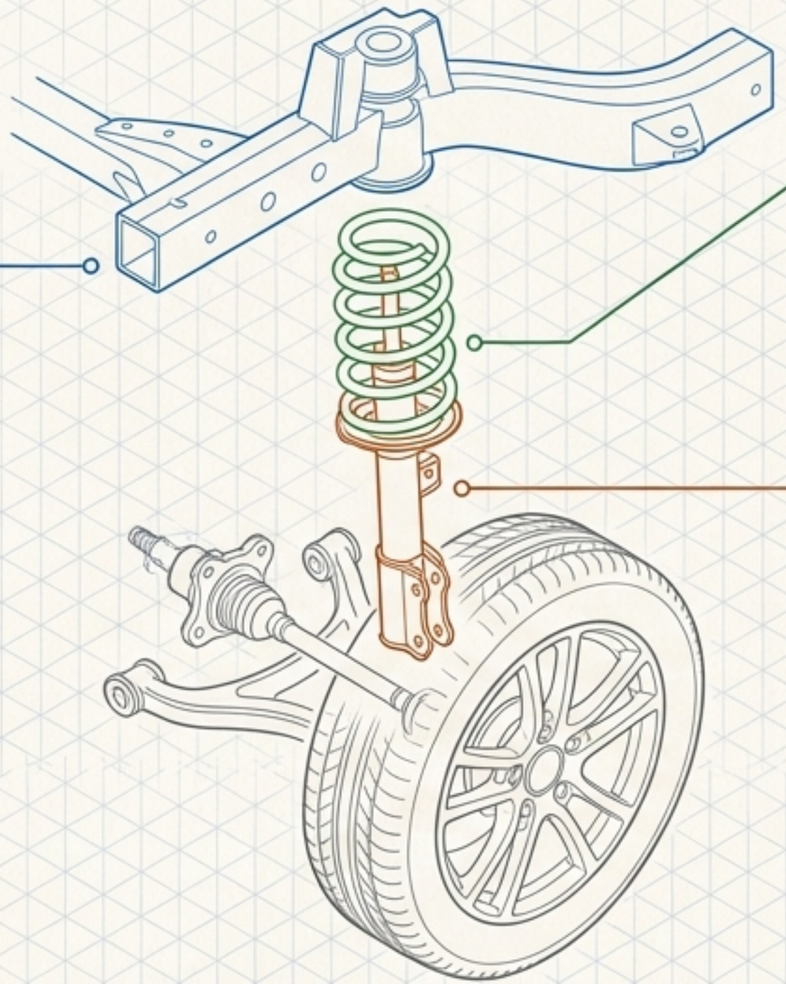
NotebookLM

The Virtual Prototyping Loop



NotebookLM

The Anatomy of a Dynamic System



The Car Body (Mass)

- **Function:** Stores Kinetic Energy.
- **Role:** The inertial resistance to acceleration.

The Suspension Spring (Spring)

- **Function:** Stores & Returns Potential Energy.
- **Role:** Creates bouncing motion; force is proportional to deflection.

The Shock Absorber (Damper)

- **Function:** Dissipates Energy as Heat.
- **Role:** Prevents continuous bouncing; generates force proportional to velocity difference.

Without damping, hitting a bump would cause the mass to bounce indefinitely on the spring. The interplay of these three fundamental elements dictates the entire system response.

NotebookLM

The Mechanical Element Dictionary

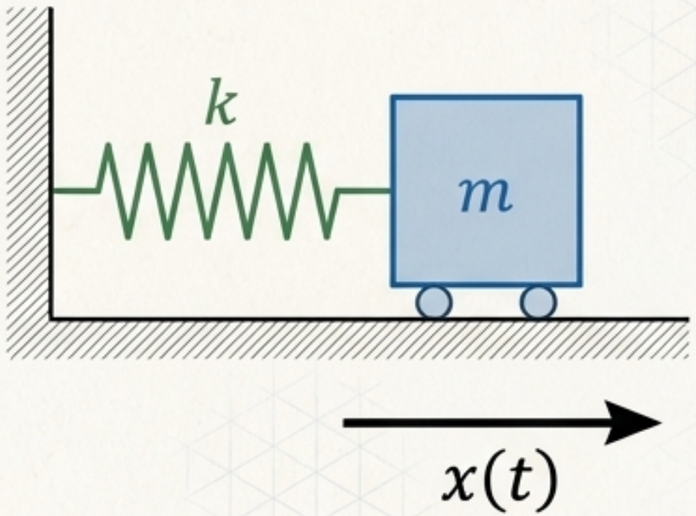
Element	Physical Function	Translational Law (Force)	Rotational Law (Torque)
Mass / Inertia	Resists motion changes.	$F = m(d^2\dot{x}/dt^{\ddot{2}})$	$M = I(d^2\ddot{\theta}/dt^{\ddot{2}})$ where $I_O = I_G + md^2$
Spring	Deflection resistance.	$F = k(x_1 - x_2)$	$T = k_t(\theta_1 - \theta_2)$
Damper	Velocity resistance.	$F = c\left(\frac{d\dot{x}_1}{dt} - \frac{d\dot{x}_2}{dt}\right)$	$T = c_t\left(\frac{d\dot{\theta}_1}{dt} - \frac{d\dot{\theta}_2}{dt}\right)$

NotebookLM

Defining the Coordinate Space (Degrees of Freedom)

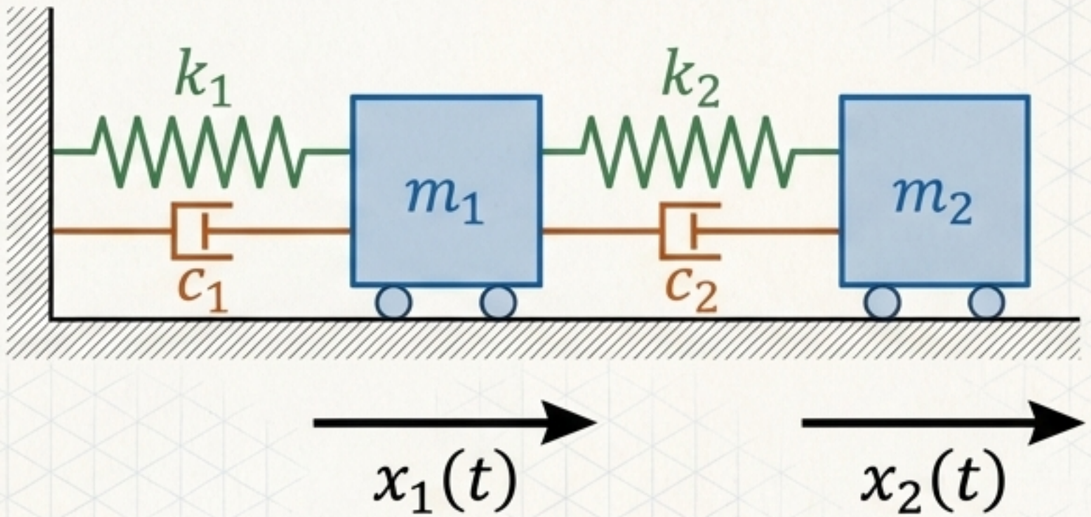
Degree of Freedom (DOF) = The number of independent generalized coordinates required to completely specify the configuration of a system.

1-DOF System



Requires only 1 coordinate to know exactly where the entire system is.

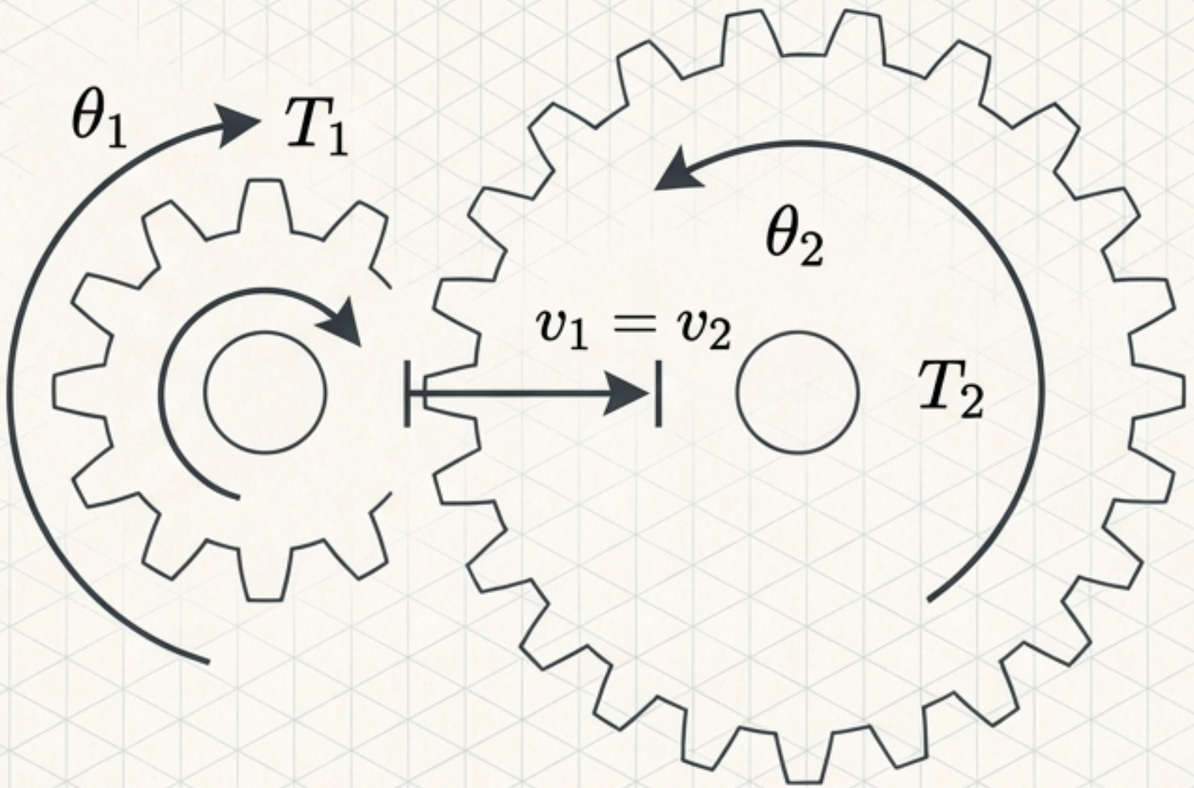
2-DOF System



The position of m_2 cannot be known merely by knowing m_1 Two independent coordinates are strictly required.

NotebookLM

Rotational Transmission: Gear Logic



Speed Ratio (n)

$$n = \frac{\theta_1}{\theta_2} = \frac{T_2}{T_1}$$

The Golden Rule

The larger gear rotates slower but outputs proportionally higher torque.

Equivalent Inertia Translation

$$J_{eq} = J_1 + \frac{J_2}{n^2}$$

Shows how a secondary inertia is ‘felt’ by the primary shaft, scaled inversely by the square of the speed ratio.

NotebookLM

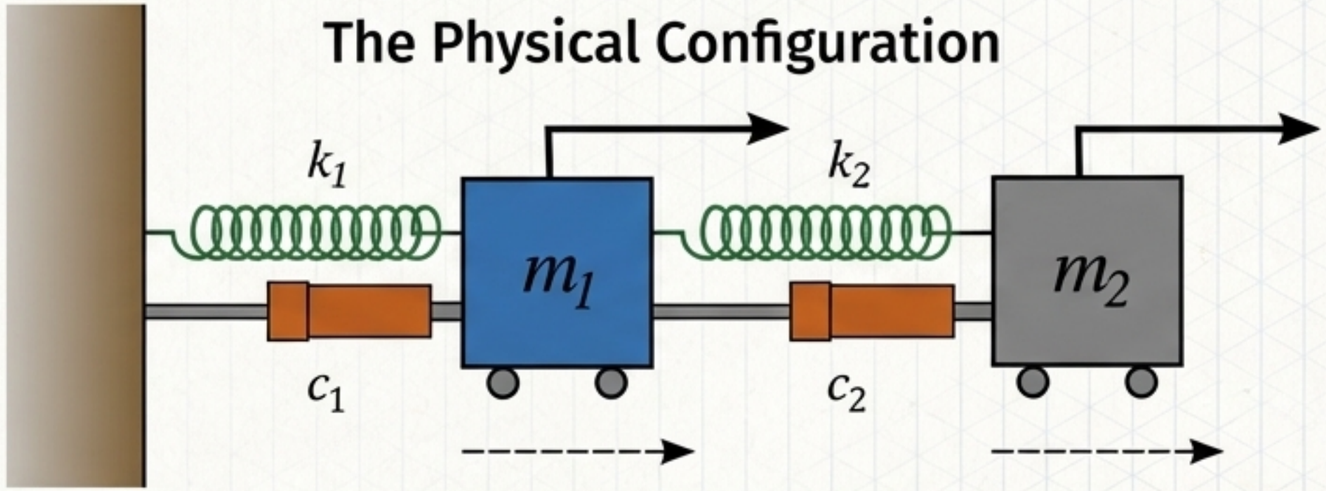
The Lenses of Dynamics: Four Mathematical Architectures

Configuration Form	State-Space Representation	Input-Output Equation	Transfer Function
$M\frac{d^2q}{dt^2} + C\frac{dq}{dt} + Kq = F$ (Matrix Form)	$\frac{dx}{dt} = Ax + Bu$ (State) $y = Cx + Du$ (Output)	Single n-th order differential equation.	$G(s) = \frac{Y(s)}{U(s)}$
Standard second-order matrix. Directly maps to physical geometry.	First-order differential equations. Ideal for multi-input, multi-output (MIMO) modern control theory.	Classical time-domain analysis.	Laplace domain. Assumes zero initial conditions. Perfect for frequency-domain analysis.

NotebookLM

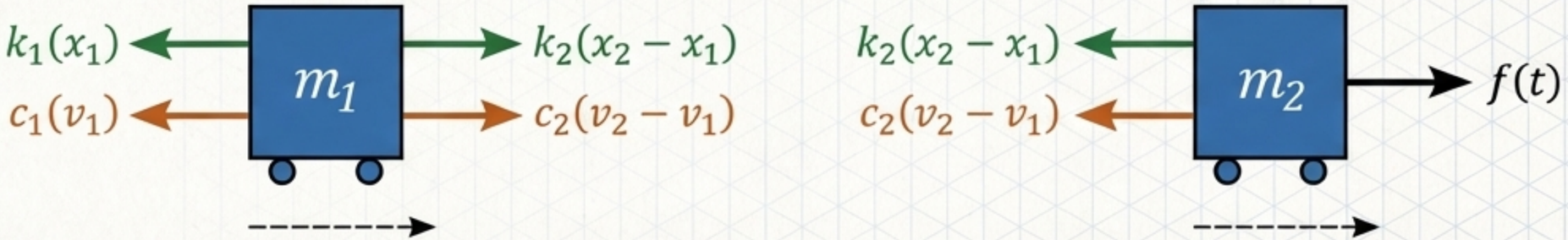
Constructing the Blueprint: Step 1 (Isolation)

The Physical Configuration



Apply Newton's 2nd Law & Isolate Elements

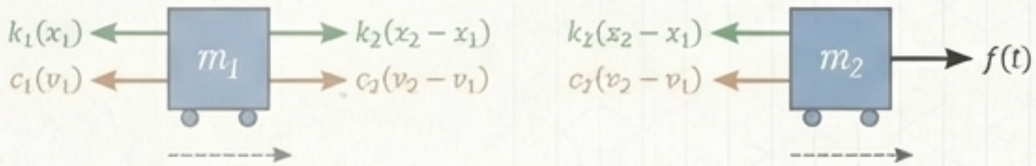
The Free Body Diagram (FBD)



NotebookLM

Constructing the Blueprint: Step 2 (Equations of Motion)

Reference Context



The Math

$$m_1\ddot{x}_1 = k_2(x_2 - x_1) + c_2(\dot{x}_2 - \dot{x}_1) - k_1x_1 - c_1\dot{x}_1$$

$$m_2\ddot{x}_2 = -k_2(x_2 - x_1) - c_2(\dot{x}_2 - \dot{x}_1) + f(t)$$

Key Insight: Every arrow on the Free Body Diagram becomes exactly one term in the differential equations. Notice how relative motion $(x_2 - x_1)$ drives the internal forces.

NotebookLM

Constructing the Blueprint: Step 3 (Configuration Form)

Packaging raw equations into the standard second-order matrix: $M(d^2q/dt^2) + C(dq/dt) + Kq = F$ in Inter

Mass/Inertia **Dampers/Dissipation** **Springs/Stiffness**

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} (x_1)'' \\ (x_2)'' \end{bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \begin{bmatrix} (x_1)' \\ (x_2)' \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ f(t) \end{bmatrix}$$

Physical coupling results in symmetric off-diagonal terms.

NotebookLM

Constructing the Blueprint: Step 4 (State-Space Architecture)

Converting n-degrees of freedom into 2n first-order equations.

Define the State Variables (**z**)

- z_1 = x_1 (Position 1)
- z_2 = (x_1)’ (Velocity 1)
- z_3 = x_2 (Position 2)
- z_4 = (x_2)’ (Velocity 2)

The State Equation Form: **dz/dt = Az + Bu**

$$\begin{bmatrix} z_1' \\ z_2' \\ z_3' \\ z_4' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{(k_1+k_2)}{m_1} & -\frac{(c_1+c_2)}{m_1} & \frac{k_2}{m_1} & \frac{c_2}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & \frac{c_2}{m_2} & -\frac{k_2}{m_2} & -\frac{c_2}{m_2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_2} \end{bmatrix} [f(t)]$$

NotebookLM

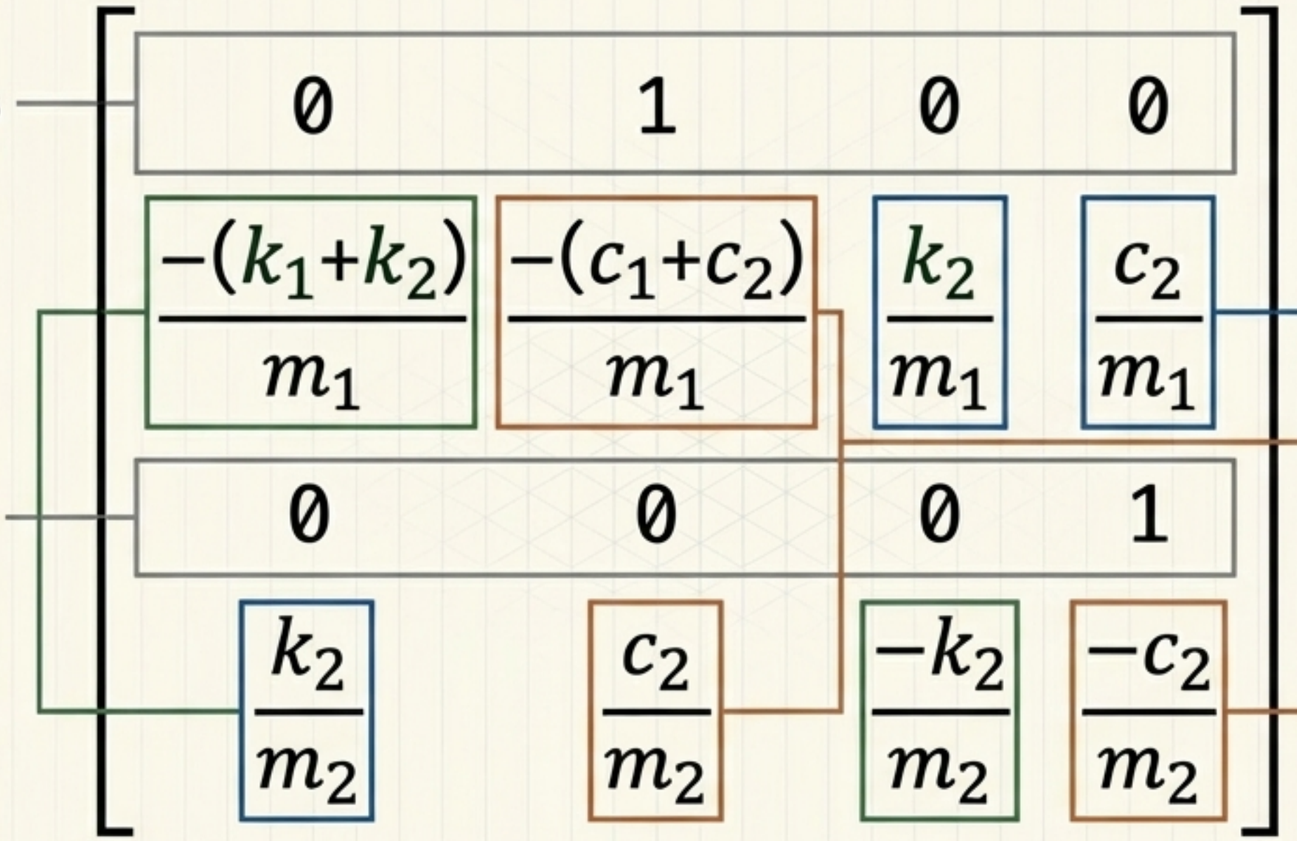
Anatomy of the System Matrix (A)

The Identity Blocks

Kinematics: Simply states that the derivative of position is velocity (z_1' = z_2).

Stiffness Map

Determines how positions generate acceleration.



Inertial Scaling

Every force is divided by mass, fulfilling **a = F/m**.

Damping Map

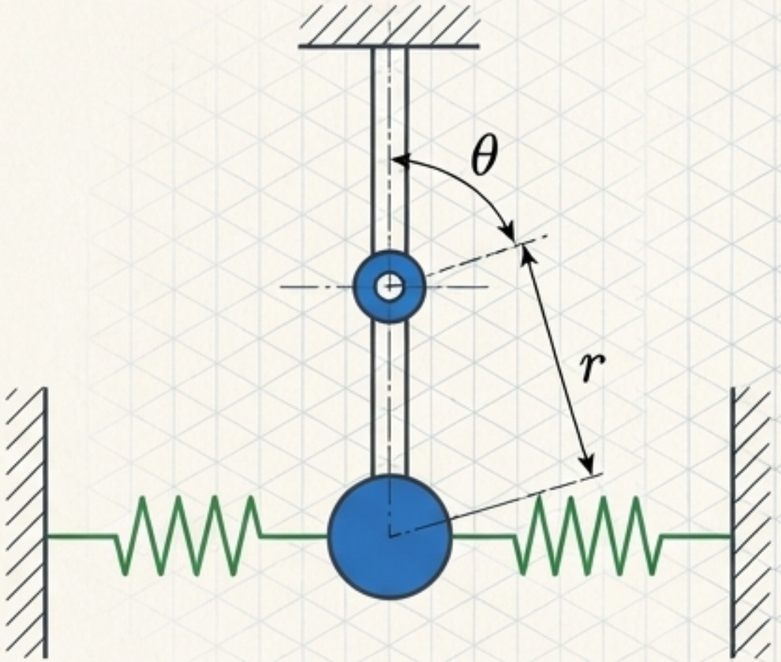
Determines how velocities generate acceleration.

Core Takeaway: A matrix is not just math; it is a perfectly organized, highly structured container for physical geometry.

NotebookLM

From Matrix to Insight: System Response

The Rotational System



By applying the matrix translation to this rotational system, we derive its core equation of motion:

$$(J_G + mr^2)\theta'' + 4kr^2\theta = 0$$

The Output Insight (Natural Frequency)

$$\omega_n = \sqrt{\frac{4kr^2}{J_G + mr^2}}$$

The model yields the natural frequency—the exact 'heartbeat' of the system. We have successfully translated a complex physical assembly into a single, predictable parameter ready for control design.

NotebookLM