Lecture (I): Robot dynamics & model-based Control

Perception
Large Language model
Athletic Intelligence
— Control System

Atlas (Boston dynamics) → No ML ✗

ANYmal → ML ✓

Manipulation hand (by openai) → ML ✓

⇒ All 3 examples started

(I) by building a detailed simulation (model)

(II) formulate control as optimization (optimization-based Controller)

→ Atlas – optimization is over parameters of a trajectory.

→ ANYmal / Hand – optimization over parameters of a neural network.

Atlas – optimization uses gradients ("differentiable simulator")

ANYmal/Hand – Just use "black-box" (no gradients)

✓ Nonlinear dynamics
✓ Optimization
✓ Control
✓ ML/RL

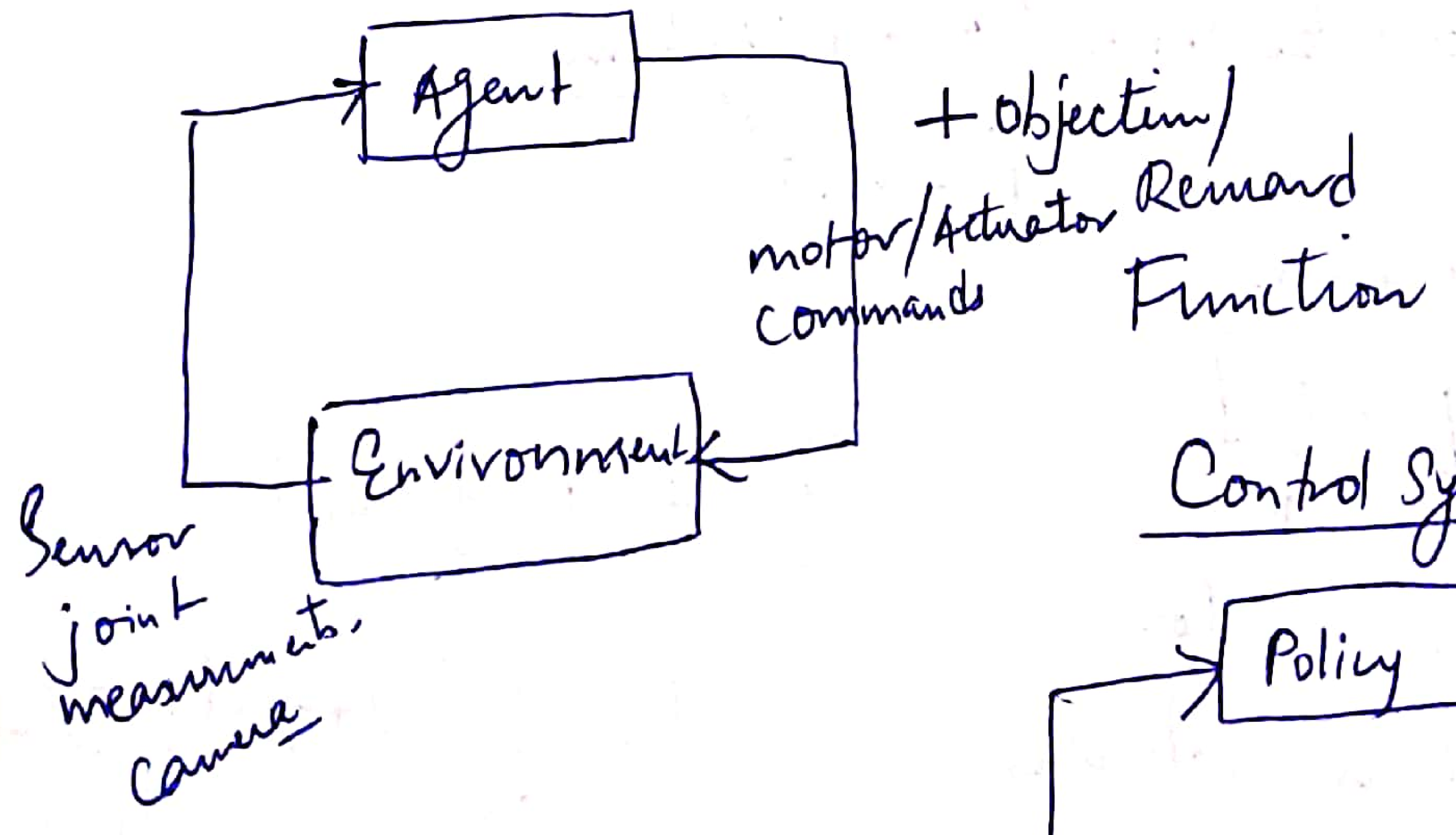
ANYmal — a quadruped robot / The quadrupedal robot

Atlas – A humanoid robot (optimization on fly / online)

(openAI – Learning Dexterous In-hand manipulation)

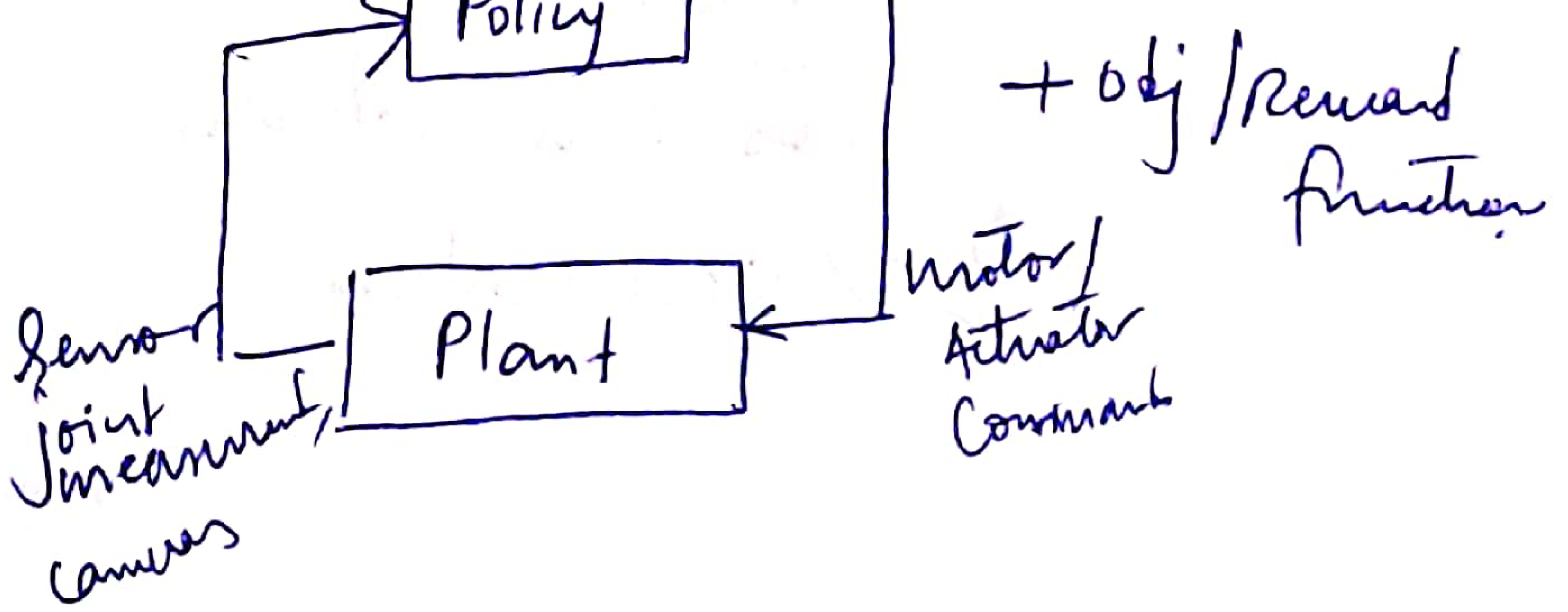
Definition:

RL



Motor is a special type of an actuator.
→ An actuator can include hydraulics, pneumatics, motor, etc.
→ Actuator is a broader term

Control System

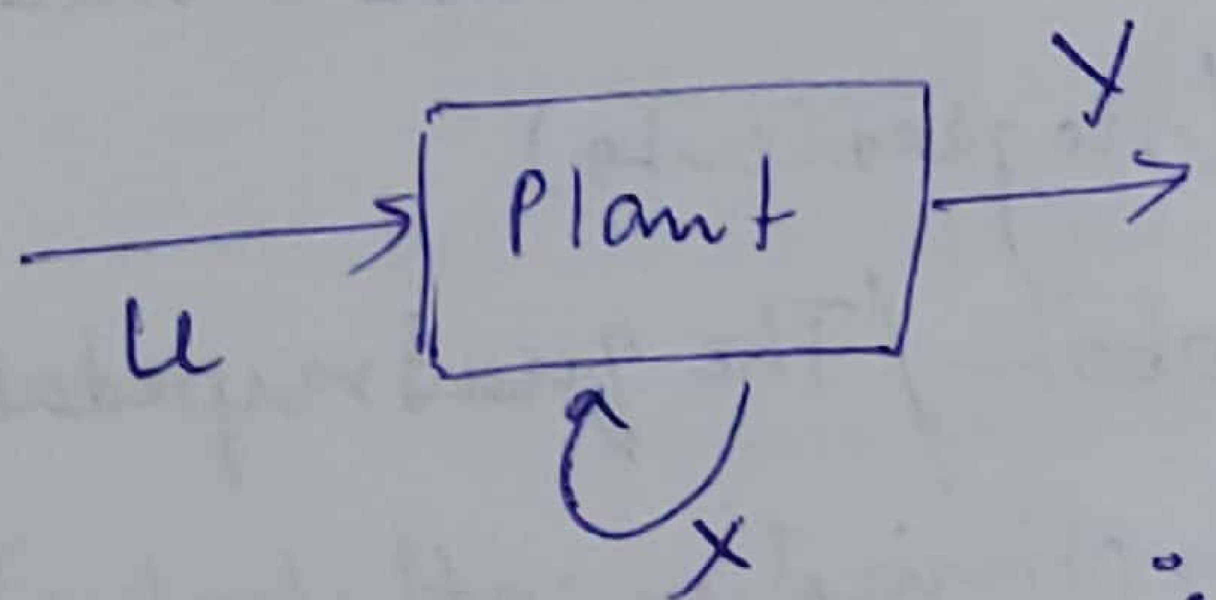


What makes Control hard?

- Uncertainty / Partial Observability
- Long-Horizon Problem (Long-term consequences of your control decisions)
- Non-linear dynamics / High-dimensional

"**Underactuated**" ⇒ implies it is one of these long-horizon problems.

The essence of the topic is – if we have a dynamical system which is underactuated then we are in a situation where control is harder because we have to think about the long-term consequences of our actions.



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

State vector

control inputs

(Vector differential eqn)

(potentially non-linear)

$\dot{x} = \frac{d}{dt} x$

$$\boxed{y = g(x)}$$

output vector (sensors)

→ Most robots are governed by lagrangian mechanics and so they tend to be <u>second order systems.</u>
($f = ma$)

For second order systems,

$$x = [q, \dot{q}]$$

q: vector of positions; $\dot{q}$: vector of velocities

$\ddot{q}$ ⇒ acceleration

$$\ddot{q} = f(q, \dot{q}, u)$$

$$\ddot{q} = f_1(q, \dot{q}) + f_2(q, \dot{q}) u$$

(affine in u or w.r.t. u)

(Control-affine 2nd order diff eqn)

<u>Note</u>: linear + constant = affine function

Definition: A system is fully-actuated if $f_2(q, \dot{q})$ is full row-rank.

Definition: A system is under-actuated if $\text{rank}\left[f_2(q, \dot{q}_2)\right] < \dim(q)$

Rank of matrix decides where control is hard or not.

Feedback Equivalence

Given q, $\dot{q}$, desired $\ddot{q}^{d}$ ← desired, then

if we were to choose $u = f_2^{-1}(q, \dot{q}) \cdot \left[\ddot{q}^d - f_1(q, \dot{q})\right]$

then this implies

$\Rightarrow \ddot{q} = \ddot{q}^d$ (only condition is f_2 is invertible)

→ Fully-actuated systems are "feedback equivalent" to the system $\ddot{q} = u$ (which is a sort of a trivial diagonal linear system; it's still 2nd order but we know a lot about it & hence how to control it).

→ ~~[illegible]~~ when it's not invertible (in case of under-actuated system) ~~[illegible]~~, then we have to do more in order to achieve some long term objectives.

→ (optimization view) — If the system is fully-actuated then we know a change of variables ~~[illegible]~~ that makes the optimization lanscape easy or convex for most objectives. There exists a simple change of variables that uses above inverse that makes the optimization lanscape easy. If it's not invertible, then we have to do more work.

Manipulator Eqns: $\ddot{q} = f_1(q,\dot{q}) + f_2(q,\dot{q})u$

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau_g(q) + Bu$$

control input

mass matrix

$KE = \frac{1}{2}\dot{q}^T M(q)\dot{q}$

Coriolis matrix

gravity vector (spring & other potential terms like in here)

actuation (selection matrix) matrix (maps our control input into q-space)

$$\boxed{\ddot{q} = \underbrace{M^{-1}(q)[}_{f_2(q,\dot{q})} Bu + \tau_g(q) - C(q,\dot{q})\dot{q}]}$$

Positive definite, symmetric.

Why does mass Matrix $M(q)$ take in 'q'?

→ ~~[illegible]~~ the effect that mass has on different joints is a configuration dependent — State dependent quantity. (it does not depend on velocity, but is state dependent)

→ Model Underactuated Systems (Acrobots, Cart-poles, Quadrotors)

→ Double pendulum

More interesting geometry/obstacles

Walking Robots / Legged robots, Simple → quadruped/ humanoids manipulator.

Stochastic/ Partially observable

Difficult to model like hydrolics, aerodynamics, etc.