The Flatness Principle: A Paradigm Shift in Nonlinear Control

Transforming Complex Dynamics into Linear Simplicity

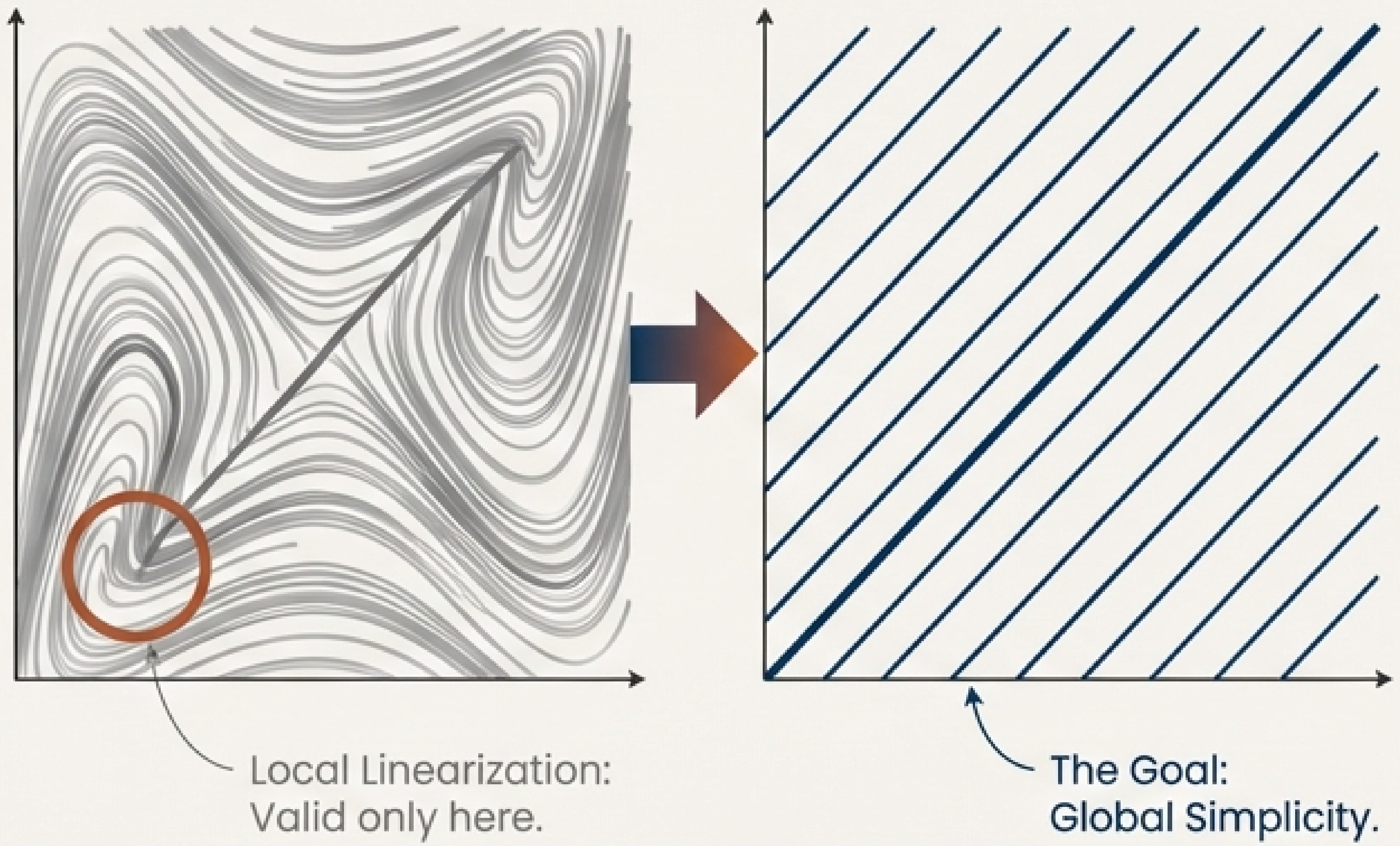
From the lecture notes on Dynamical Systems, SS 2025, Technical University of Munich

Why Nonlinear Control Is So Hard

- **Complex Dynamics**
  Trajectory planning and tracking for general nonlinear systems ($\dot{x} = f(x, u)$) require solving complex differential equations, which is often computationally intensive and lacks analytical solutions.

- **Local vs. Global Validity**
  Standard linearization via Taylor series approximation is only valid in a small neighborhood around a reference trajectory or an equilibrium point. The linear model loses accuracy as the system moves away from this point.

- **Hidden Internal Dynamics**
  Partial linearizations (like I/O linearization) can leave behind "internal dynamics" or "zero dynamics." If these are unstable, the entire system can become unstable, even if the output appears controlled.



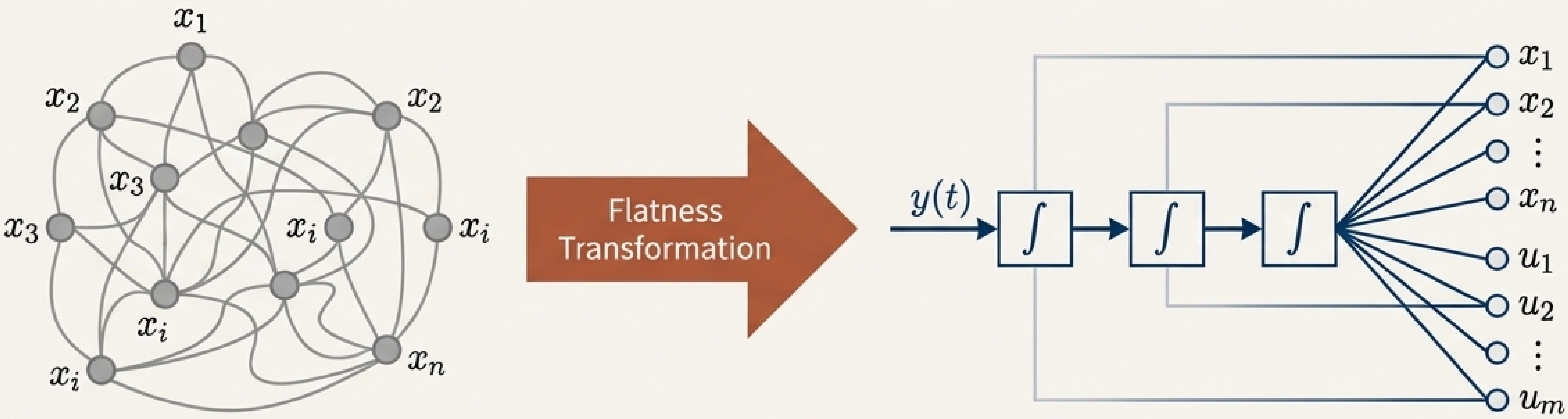
How can we design controllers that are exact, predictable, and robust over a wide operating range without the limitations of local approximations?

What If We Could Find a ‘Golden Variable’ to Describe Everything?

The Core Concept: Differential Flatness

Flatness is a property of certain nonlinear systems that allows their entire dynamic behavior—all states and all necessary inputs—to be parameterized by a special set of outputs and their time derivatives.

This special output is called the **flat output** (#003366). It’s not necessarily a physical measurement but a mathematical construct.



Complex Nonlinear System

Flat Output Representation

The Key Insight

For a flat system, we don’t approximate. We perform an **exact transformation** (#003366) of the complex nonlinear dynamics into a simple, linear representation (a chain of integrators).

The Flat Output: A Variable That Uniquely Defines the System Trajectory

A system $\dot{x} = f(x, u)$ is differentially flat if a fictitious output y exists, where $\dim(y) = \dim(u)$, such that:

$y(t)$ and its time derivatives

1. All states x are functions of y and its finite time derivatives.

$$x = \Psi_1\left(y, \dot{y}, \ldots, y^{(\gamma-1)}\right)$$

2. All inputs u are also functions of y and its finite time derivatives.

$$u = \Psi_2\left(y, \dot{y}, \ldots, y^{(\gamma)}\right)$$

System State $x(t)$

Control Input $u(t)$

Implication: The entire system trajectory, both $x(t)$ and $u(t)$, is completely and uniquely determined by the trajectory of the flat output $y(t)$. The components of y are differentially independent.

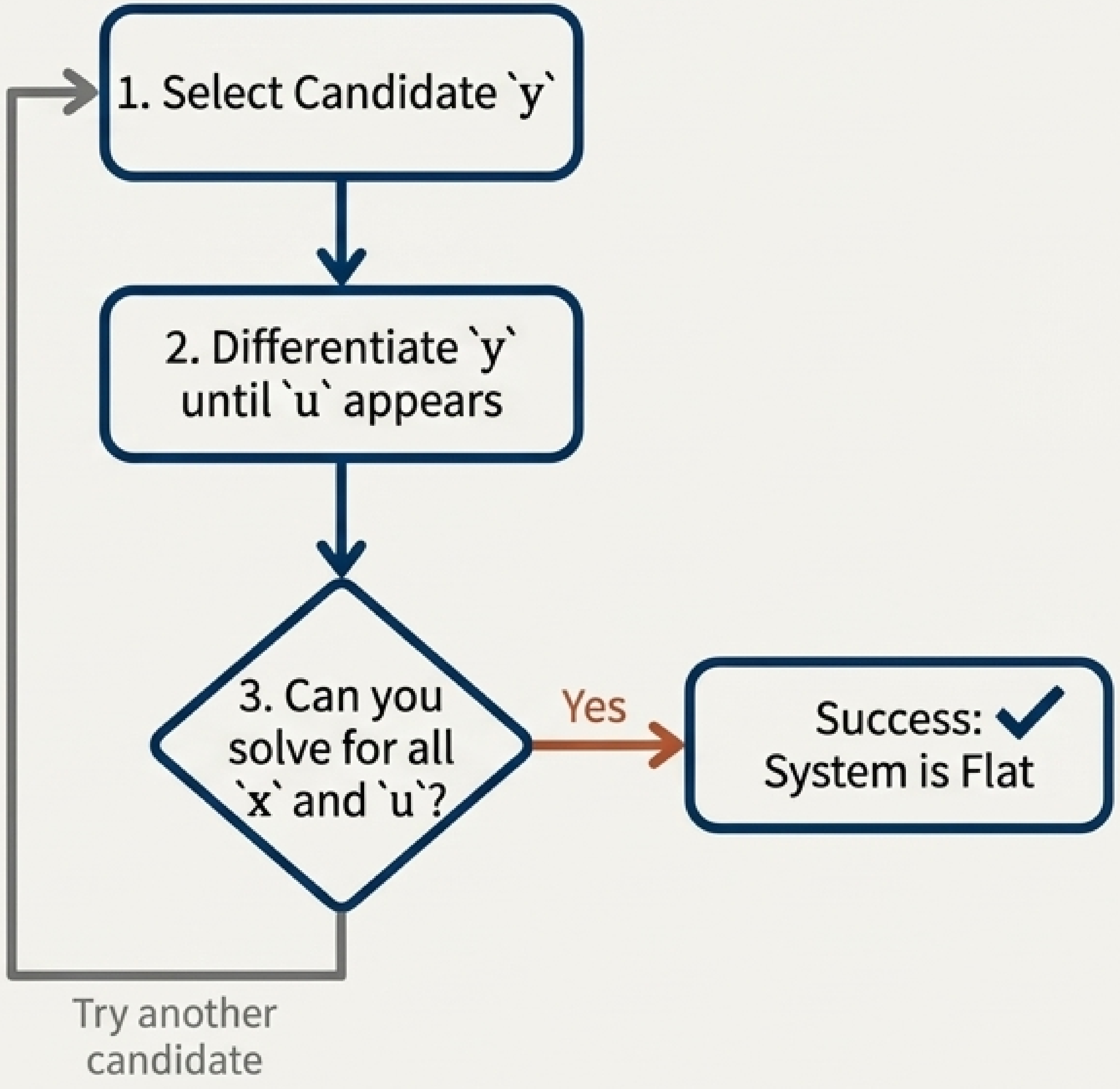
The Critical Task: How to Find the Flat Output

The Challenge

There is no universal algorithm for finding a flat output. It oftonn requires physical insight into the system.

A Heuristic Procedure

1. **Select a Candidate \`y\`:** Choose a potential output that seems to capture the essential motion or information of the system. \`dim(y)\` must equal \`dim(u)\`.
2. **Differentiate Successively:** Calculate the time derivatives \`$\dot{y}, \ddot{y}, \ldots$\` using the system's state equations $\dot{x} = f(x, u)$.
3. **Check for Invertibility:** At each step, check if the resulting system of algebraic equations can be inverted to express all states \`x\` and inputs \`u\` in terms of \`y\` and its derivatives alone.
4. **Confirm:** If all states and inputs can be uniquely determined, the candidate \`y\` is a flat output.



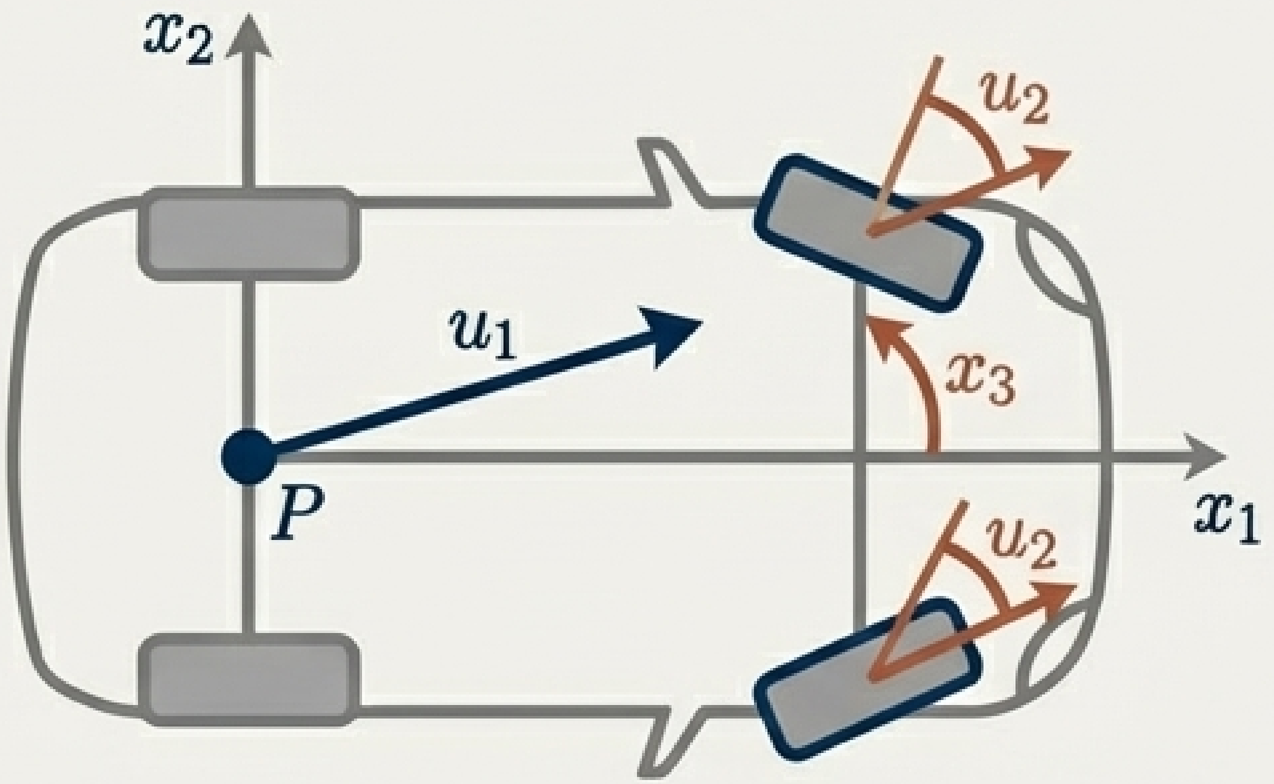
NotebookLM

Case Study, Part 1: Flatness Analysis of a Kinematic Vehicle

System Model

$\dot{x}_1 = u_1 \cos(x_3)$
$\dot{x}_2 = u_1 \sin(x_3)$
$\dot{x}_3 = \frac{u_1}{L} \tan(u_2)$ (with L = 1 for simplicity).

States x: $[x_1, x_2]$ = position, x_3 = heading angle
Inputs u: u_1 = velocity, u_2 = steering angle



Analysis

1. **Candidate Flat Output:** We select the position of the rear axle center point: $y = [y_1, y_2]^T = [x_1, x_2]^T$. Note $\dim(\mathbf{y}) = 2$, $\dim(u) = 2$.
2. **First Derivatives:**
   $\dot{y}_1 = \dot{x}_1 = u_1 \cos(x_3), \dot{y}_2 = \dot{x}_2 = u_1 \sin(x_3)$
3. **Solving for States and Inputs:** From the first derivatives, we can algebraically solve for state x_3 and input u_1:
   $$x_3 = \arctan\left(\frac{\dot{y}_2}{\dot{y}_1}\right), \; u_1 = \frac{\dot{y}_1}{\cos(x_3)} = \sqrt{\dot{y}_1^2 + \dot{y}_2^2}$$
4. **Solving for the Remaining Input:** By differentiating further $(\ddot{y}_1, \ddot{y}_2)$ and substituting, we can also solve for the second input u_2 in terms of $y, \dot{y}, \ddot{y}$.

Conclusion: Since all states (x_1, x_2, x_3) and all inputs (u_1, u_2) can be expressed as functions of y and its derivatives, the system is differentially flat, and $y = [x_1, x_2]^T$ is a valid flat output.

Payoff I: Effortless Feedforward Control

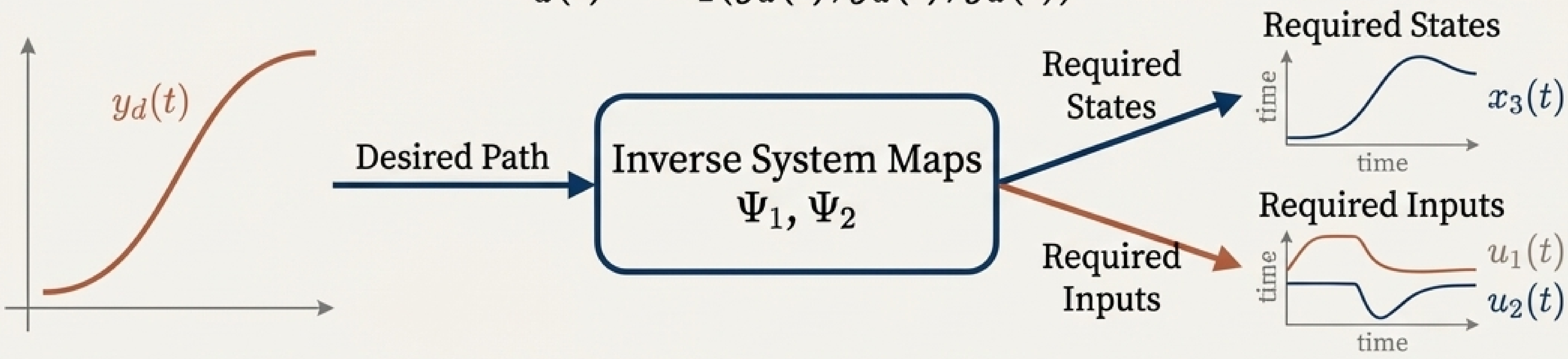
The Trajectory Generation Problem

Given a desired path for the vehicle, $y_d(t) = [x_{1d}(t), x_{2d}(t)]^T$, what are the exact velocity and steering inputs, $u_d(t)$, required to follow this path perfectly?

The Flatness Solution

Because we have the inverse map $u = \Psi_2(y, \dot{y}, \ddot{y}, \dots)$, we can directly calculate the required nominal control input. Simply differentiate the desired path $y_d(t)$ and substitute these trajectories directly into the functions derived during the flatness analysis:

$$u_d(t) = \Psi_2(y_d(t), \dot{y}_d(t), \ddot{y}_d(t))$$



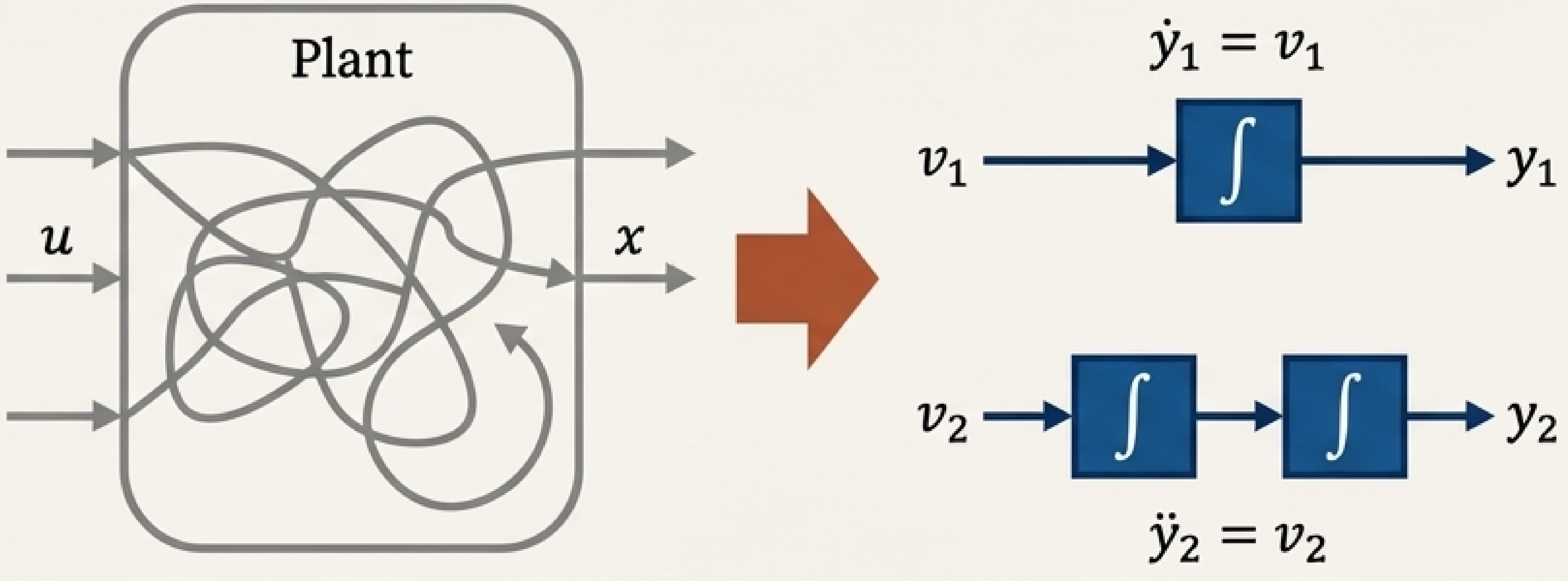
The Result: This calculation provides the exact open-loop (feedforward) control $u_d(t)$ needed to follow the trajectory, assuming a perfect model and no disturbances.

Payoff II: Exact Feedback Linearization for Robust Tracking

The Need for Feedback
Feedforward control is not robust to model errors or disturbances. We need feedback to drive tracking errors to zero.

The Flatness Advantage
We can design a feedback law that results in **exactly linear error dynamics.** This is not an approximation.

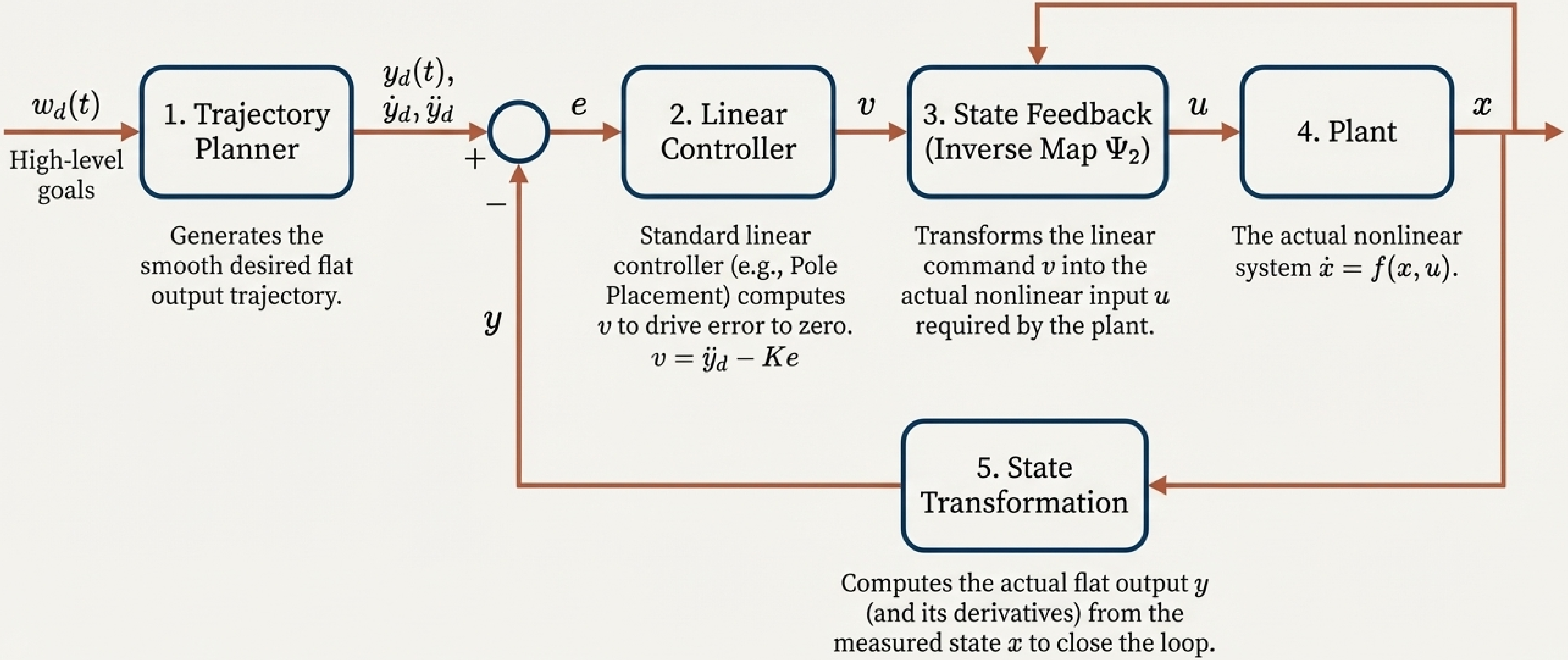
The Method
1. **Define New Coordinates:** Create a new state z from the flat output and its derivatives, e.g., $z = [y_1, y_2, \dot{y}_2]^T$.
2. **Define Linearizing Inputs:** Define new inputs v as the highest-order derivatives of y, e.g., $v_1 = \dot{y}_1, v_2 = \ddot{y}_2$.
3. **The Transformation:** The system in new (z, v) coordinates becomes a set of simple, decoupled integrator chains.
4. **Design a Linear Controller:** Use a simple linear controller (e.g., PD) for v to stabilize the tracking error $e = y - y_d$. Example: $v_2 = \ddot{y}_{2_d} + p_2(\dot{y}_{2_d} - \dot{y}_2) + p_3(y_{2_d} - y_2)$



Original Nonlinear System

Transformed Flat System

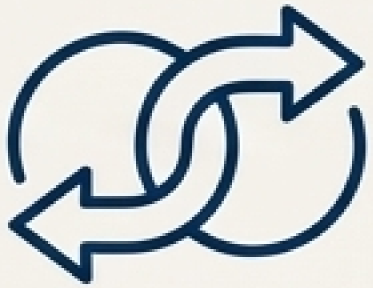
The Complete Flatness-Based Control Architecture



Fundamental Properties and Implications of Flatness

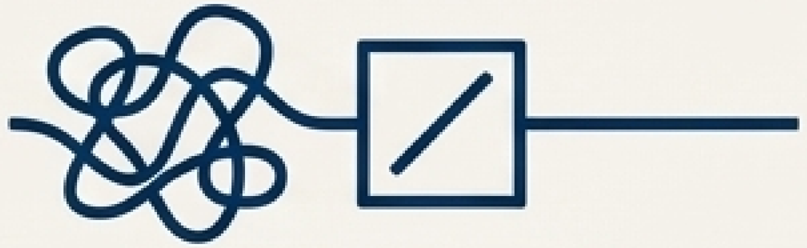
Controllability

Every flat system is controllable. The flat output provides a direct "handle" to steer the system's entire state trajectory.



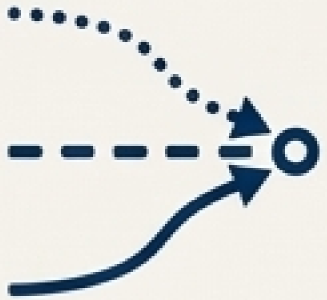
System Inversion

Flatness is fundamentally equivalent to the existence of a (dynamic) inverse system. The feedforward controller is a direct implementation of this inverse.



Powerful Linearization

Flat systems are exactly linearizable via *dynamic* state feedback. This is a more general and powerful concept than static feedback linearization.



Non-Uniqueness and Flexibility

A flat output for a given system is not unique. This provides design freedom to choose an output that may be physically meaningful or computationally convenient.

The Flatness Principle: Key Takeaways

Summary of the Core Idea

- **The Transformation**: Flatness is a property that allows transforming a complex nonlinear control problem into a simple linear one by parameterizing the system with a 'flat output' y.
- **The Process**: The design involves two key stages: (1) Heuristic analysis to find the flat output and derive the system's inverse maps. (2) Designing control based on this new representation.

The Two Powerful Payoffs

- **Payoff 1: Feedforward Control**: Trajectory generation becomes a simple algebraic substitution, providing the exact inputs to follow any desired path $y_d(t)$.
- **Payoff 2: Feedback Control**: The system's error dynamics can be made **exactly linear**, enabling robust, predictable tracking performance with standard linear control techniques.