# *ECE 6552 – Lecture 14*

## MOTIVATING FEEDBACK LINEARIZATION

*March 3 2026*

Overview:

- Control Systems and Control Objectives
- Motivating Feedback Linearization

Additional Reading

- Khalil Chapter 12 (Feedback Control)
- Khalil Chapter 13 (Feedback Linearization)

## *Course Roadmap*

Proposed Roadmap:

- Lecture 14-18: Feedback Linearization
- Lecture 19-21: Control Lyapunov Functions
- Lecture 22-24: Control Barrier Functions
- Lecture 25: Exam 2 Review
- Lecture 26: Hybrid Zero Dynamics

## *Motivating Feedback Linearization*

Feedback linearization is an extremely powerful tool in nonlinear control because it allows one (under certain conditions) to *exactly* linearize a nonlinear system via nonlinear control.

### *Control Systems and Control Objectives*

**Definition**: *Control System.* Consider an open and connected set $D \subseteq \mathbb{R}^n$ and a set of admissible control inputs $U \subset \mathbb{R}^m$. A control system is given by a differential equation:

$$\dot{x} = f(x, u), \quad x \in D, u \in U$$

where $f : D \times U \to \mathbb{R}^n$ is a $C^1$ function

**Definition**: *Affine Control System.* A control system is an affine control system (or control-affine) if it can be written in the form:

$$\dot{x} = f(x) + g(x)u$$

where $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^{n\times m}$ are $C^1$ functions. Here, $f$ is sometimes called the *drift* and $g$ is sometimes called the *actuation matrix* or *input matrix*.

**Definition**: *Outputs.* An output is a differentiable function $y : D \to R^k$, sometimes written in vector form:

$$y(x) = \begin{bmatrix} y_1(x) \\ \vdots \\ y_k(x) \end{bmatrix} \in \mathbb{R}^k$$

**Goal of Control:** Control objectives can be mathematically encoded through the use of *outputs*. Explicitly, the goal is to define a feedback control law $u : D \to U$ such that any solution $x(t)$ of the resulting closed loop system:

$$\dot{x} = f_{cl}(x) = f(x) + g(x)u(x)$$

drives the outputs to zero (drives $y(x(t)) \to 0$ as $t \to \infty$).

**Example:** One standard control objective is to drive the system to a desired state $x_{\text{des}} \in \mathbb{R}^n$. In this case:

$$y(x) = x - x_{\text{des}}$$

where $y : D \to \mathbb{R}^n$ and thus $k = n$.

**Virtual Constraints:** We can achieve "output-based tracking" by defining a set of *virtual constraints*:

$$y(x,t) = y_a(x) - y_{\text{des}}(t,\alpha)$$

where $y_a(x)$ is the actual output and $y_{\text{des}}(t,\alpha)$ is the desired output, which is a function of time and some parameterization $\alpha$. We can remove the dependence on time using a *parameterization of time* $\tau : D \to \mathbb{R}$. This allows us to represent our virtual constraints as:

$$y(x) = y_a(x) - y_{\text{des}}(\tau(x),\alpha)$$

Therefore, achieving the objective $y \to 0$ as $t \to \infty$ implies that $y_a \to y_{\text{des}}$ as $t \to \infty$.

**Beziér Polynomials:** In the context of robotic systems, it is often useful to parameterize desired motions using a polynomial parameterization. One of the most popular choices is Beziér polynomials[1]. A Beziér polynomial of degree $M$ is defined as:

$$\begin{aligned} y_d(t,\alpha)_i &= \sum_{k=0}^{M} \frac{M!}{k!(M-k)!} \alpha_{k,i} t^k (1-t)^{M-k} \\ &= \sum_{k=0}^{M} \underbrace{\binom{M}{k}}_{\text{binom.}} \underbrace{\alpha_{k,i}}_{\text{coeff.}} \underbrace{t^k (1-t)^{M-k}}_{\text{polynom. term}} \end{aligned}$$

[1] For a great interactive tutorial on Beziér polynomails see https://pomax.github.io/bezierinfo/

where $\alpha_{k,i}$ are called the control points of the Beziér curve.

**Parameterization of Time**: In the case of robotic walking, we can parameterize time as the forward evolution of a walking robot through a step. Explicitly, this is done by defining a function $\tau : D \to \mathbb{R}$:

$$\tau(x) = \frac{\theta(q) - \theta^+}{\theta^- - \theta^+}$$

where $\theta : D \to \mathbb{R}$ is a phase variable quantifying the forward progression of the robot (it must be monotonic), $\theta^+ = \theta(x^+)$ is its value at the "beginning" of the step, and $\theta^- = \theta(x^-)$ is its value at the "end" of the step. One common choice for $\theta$ is the angle of the stance leg with respect to the vertical axis.

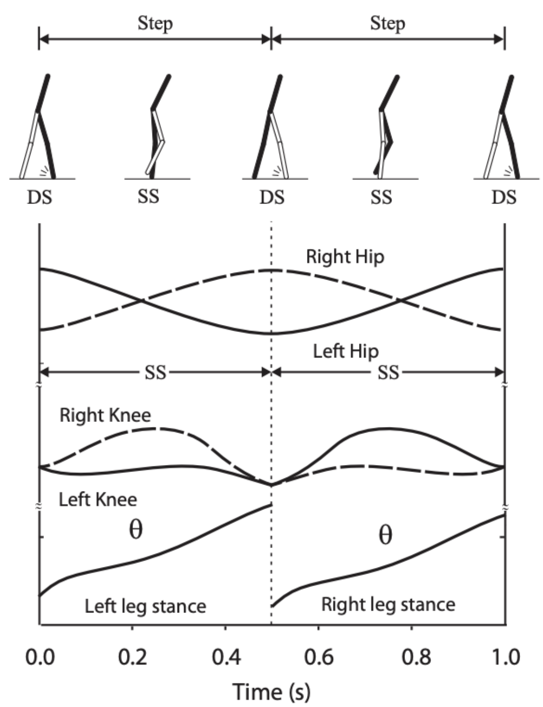


Figure 1: Figure 6 of [Grizzle et al., 2014]

*Feedback Linearization*

We will begin by considering SISO (single input single output) systems. These are systems with $k = m = 1$ wherein the system takes the form:

$$\dot{x} = f(x) + g(x)u$$
$$y = h(x)$$

with $x \in D \subseteq R^n$, $u \in U \subset \mathbb{R}$, and $h : D \to \mathbb{R}$.

**Definition**: *Feedback Linearizable.* A control system is feedback linearizable (or input-state linearizable) if there exists a diffeomorphism $z = T(x)$ and a feedback control law $u : D \times U \to U$ (i.e., $u(x, v)$) such that the closed loop system:

$$\dot{x} = f(x) + g(x)u(x, v)$$

with $v \in \mathbb{R}$ being a new control input, renders a linear relationship between the input and the state:

$$\dot{z} = Az + Bv$$

Since $v$ is an *auxiliary input*, it can be chosen to stabilize the system dynamics ($x$) which are now linear.

However, sometimes (such as the case with tracking control) it is more beneficial to linearize the input-output map rather than the input-state map. This is called *input-output linearization*.

**Definition**: *Input-Output Linearizable.* A control system is input-output linearizable if there exists a feedback control law $u : D \times U \to U$ (i.e., $u(x, v)$) such that the closed loop system:

$$\dot{x} = f(x) + g(x)u(x, v)$$

with $v \in \mathbb{R}$ being a new control input, renders a linear relationship between the input and the output:

$$y^{(p)} = v$$

with $p$ denoting the *relative degree* of the system.

In this case, since $v$ is an *auxiliary input*, it can be chosen to stabilize the output dynamics ($y$) which are now linear.

**Feedback Linearization Example** [2]: Let's consider an inverted pendulum with torque actuation at the pivot point:

[2] Example code can be found online

$$\dot{x} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin(x_1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

with $x_1 = \theta$ being the angle (measured from the downward vertical) and $x_2 = \dot{\theta}$ begin the angular velocity.

Our control objective is to drive the pendulum to the upright position $\theta = \pi$, encoded by:

$$y = h(x) = x_1 - \pi$$

This system can be feedback linearized by choosing the control law $u = \frac{g}{l}\sin(x_1) + v$. Plugging this into the system dynamics yields the closed-loop system:

$$\dot{x} = \begin{bmatrix} x_2 \\ v \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} v$$

To go further, we can select $v = -k_p y - k_d \dot{y}$ to both stabilize the closed-loop system (for the shifted system $\tilde{x} = x - (\pi, 0)$ such that the equilibrium point is at the origin) and to drive the output to zero. This results in the closed-loop system:

$$\dot{\tilde{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix}}_{A_{cl}} \tilde{x}$$

Thus, we can stabilize our system by choosing $k_p$ and $k_d$ such that the eigenvalues of $A_{cl}$ have negative real parts.

We can also analyze the behavior of the output dynamics:

$$\begin{aligned} y &= x_1 - \pi \\ \dot{y} &= \dot{x}_1 = x_2 \\ \ddot{y} &= \dot{x}_2 = v \end{aligned}$$

Thus, the same conditions can be enforced on $k_p$ and $k_d$ to stabilize the second-order output dynamics:

$$\ddot{y} + k_d \dot{y} + k_p y = 0$$

## *References*

Jessy W Grizzle, Christine Chevallereau, Ryan W Sinnet, and Aaron D Ames. Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica*, 50(8):1955–1988, 2014.