Module 1 — Feedback Loops

Systems Thinking

hbar.university

Feedback Mechanisms

Position in Knowledge Graph

This node introduces the fundamental structural object of the course: the feedback mechanism.

It precedes: - Positive and Negative Feedback (1.2) - Formal Dynamical Systems (2.1)

Feedback is the ontological primitive from which recursive dynamics arise.

Formal Statement / Definition / Construction

Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a state space.

A **feedback mechanism** is a tuple

$$\mathcal{F} = (\mathcal{X}, F)$$

where

$$F : \mathcal{X} \to \mathcal{X}$$

is a state-update mapping.

The induced discrete-time evolution is defined recursively by

$$x_{t+1} = F(x_t), \quad t \in \mathbb{N}.$$

Equivalently, a continuous-time feedback mechanism is defined by

$$\dot{x}(t) = f(x(t)),$$

where

$$f : \mathcal{X} \to \mathbb{R}^n$$

is a vector field.

In both cases, the defining structural property is:

$$\text{Output at time } t \text{ influences input at time } t+1.$$

Formally, the system is **self-referential**:

$$x_{t+1} \text{ depends on } x_t.$$

Assumptions and Conditions

1. The state space $\mathcal{X}$ is a nonempty subset of $\mathbb{R}^n$.
2. The mapping F is well-defined on $\mathcal{X}$.
3. For continuous-time systems, assume f is locally Lipschitz to guarantee local existence and uniqueness of solutions.

No assumptions are made regarding linearity, stability, or boundedness at this stage.

Proof or Mathematical Development

A feedforward system has the form

$$y = G(u),$$

where output y depends only on input u.

In contrast, a feedback system satisfies

$$u = H(y),$$

so that

$$y = G(H(y)).$$

Thus,

$$y = (G \circ H)(y).$$

Let

$$F = G \circ H.$$

Then the system reduces to a self-mapping:

$$y = F(y).$$

In discrete-time evolution,

$$x_{t+1} = F(x_t)$$

is obtained by iterating this self-mapping.

Thus, feedback induces a dynamical system through function iteration.

The structure is recursive:

$$x_{t+k} = F^k(x_t),$$

where

$$F^k = \underbrace{F \circ F \circ \cdots \circ F}_{k \text{ times}}.$$

This recursive iteration is the mathematical engine generating trajectories.

Structural Role

Feedback mechanisms introduce:

- State recursion
- Temporal dependence
- Self-reference
- Potential for amplification or regulation

All later domains—dynamics, stability, control, networks, and emergence—require the existence of recursive state evolution.

Without feedback, no trajectory exists.

Feedback is the minimal structure that produces time-evolution.

Structural Compression

Invariant

A feedback mechanism is a self-mapping:

$$F : \mathcal{X} \to \mathcal{X}$$

whose iteration defines system evolution.

Mechanism

Recursive composition:

$$x_{t+1} = F(x_t), \quad x_{t+k} = F^k(x_t).$$

Cross-System Echo

- Linear Algebra: eigenvalues govern iteration of linear maps.
- Control Theory: closed-loop systems are feedback compositions.
- Economics: expectations feedback into price formation.
- Biology: regulatory networks are state-dependent update rules.

Positive and Negative Feedback

Position in Knowledge Graph

This node refines the general feedback mechanism (1.1) by classifying feedback according to its local effect on deviations from equilibrium.

It prepares for: - Oscillations and delays (1.5) - Fixed points (2.3) - Stability analysis (3.1)

Formal Statement / Definition / Construction

Let $\mathcal{X} \subseteq \mathbb{R}$ and consider a discrete-time feedback system

$$x_{t+1} = F(x_t), \quad F : \mathcal{X} \to \mathcal{X}.$$

Let $x^* \in \mathcal{X}$ be a fixed point, i.e.

$$F(x^*) = x^*.$$

Define the deviation variable

$$\delta_t = x_t - x^*.$$

Assume F is differentiable at x^*. A first-order Taylor expansion gives

$$F(x_t) = F(x^*) + F'(x^*)(x_t - x^*) + r(x_t),$$

where

$$\lim_{x_t \to x^*} \frac{r(x_t)}{x_t - x^*} = 0.$$

Using $F(x^*) = x^*$, we obtain

$$\delta_{t+1} = F'(x^*)\, \delta_t + r(x_t).$$

Neglecting higher-order terms locally:

$$\delta_{t+1} \approx F'(x^*)\, \delta_t.$$

Definition (Local Positive Feedback)

The system exhibits **local positive feedback** at x^* if

$$|F'(x^*)| > 1.$$

Small deviations are amplified:

$$|\delta_{t+1}| > |\delta_t| \quad \text{for sufficiently small } \delta_t \neq 0.$$

Definition (Local Negative Feedback)

The system exhibits **local negative feedback** at x^* if

$$|F'(x^*)| < 1.$$

Small deviations are attenuated:

$$|\delta_{t+1}| < |\delta_t| \quad \text{for sufficiently small } \delta_t \neq 0.$$

Assumptions and Conditions

1. $\mathcal{X} \subseteq \mathbb{R}$.
2. F is differentiable in a neighborhood of x^*.
3. The analysis is local.
4. Higher-order terms are neglected only for sufficiently small deviations.

This classification is local and does not imply global stability.

Proof or Mathematical Development

From the linearized deviation equation:

$$\delta_{t+1} \approx F'(x^*)\,\delta_t.$$

Iterating,

$$\delta_t \approx \left(F'(x^*)\right)^t \delta_0.$$

Case 1: $|F'(x^*)| < 1$.

Then

$$\lim_{t\to\infty} \left(F'(x^*)\right)^t = 0,$$

so

$$\delta_t \to 0.$$

Perturbations decay.

Case 2: $|F'(x^*)| > 1$.

Then

$$\left|\left(F'(x^*)\right)^t\right| \to \infty,$$

so deviations grow.

Thus amplification vs attenuation is governed by the local gain $F'(x^*)$.

Structural Role

Positive and negative feedback determine:

- Divergence vs convergence
- Growth vs regulation
- Instability vs stabilization

All stability theory in later domains is a formalization of this local gain condition.

Positive feedback generates amplification and potential runaway behavior.

Negative feedback generates regulation and homeostasis.

This classification is the first quantitative refinement of the abstract feedback mechanism.

Structural Compression

Invariant

Local behavior near equilibrium is governed by the derivative:

$$\delta_{t+1} \approx F'(x^*)\,\delta_t.$$

Mechanism

Linearization and iteration of the feedback map determine amplification or attenuation.

Cross-System Echo

- Linear Algebra: spectral radius governs iteration of linear operators.
- Control Theory: loop gain determines closed-loop stability.
- Economics: expectation amplification vs stabilizing policy feedback.
- Biology: regulatory inhibition vs activation loops.

Causal Loop Diagrams

Position in Knowledge Graph

This node formalizes causal loop diagrams (CLDs) as graphical representations of feedback structure.

It builds on: - Feedback mechanisms (1.1) - Positive and negative feedback (1.2)

It prepares for: - Stock-and-flow modeling (1.4) - Graph-theoretic network analysis (4.1)

Formal Statement / Definition / Construction

Let $V = \{x_1, \dots, x_n\}$ be a finite set of system variables.

A **causal loop diagram (CLD)** is a signed directed graph

$$\mathcal{G} = (V, E, \sigma),$$

where:

- $E \subseteq V \times V$ is a set of directed edges,
- $\sigma : E \to \{+1, -1\}$ assigns a sign to each edge.

An edge $(x_i, x_j) \in E$ with sign σ_{ij} represents a causal relation:

$$\frac{\partial x_j}{\partial x_i} \text{ has sign } \sigma_{ij}.$$

Interpretation:

- $\sigma_{ij} = +1$: an increase in x_i tends to increase x_j (all else equal).
- $\sigma_{ij} = -1$: an increase in x_i tends to decrease x_j.

Definition (Feedback Loop)

A **feedback loop** is a directed cycle

$$x_{i_1} \to x_{i_2} \to \cdots \to x_{i_k} \to x_{i_1}$$

in $\mathcal{G}$.

The **loop sign** is defined as

$$\Sigma = \prod_{m=1}^{k} \sigma_{i_m i_{m+1}},$$

with indices taken modulo k.

- $\Sigma = +1$: reinforcing (positive) loop.
- $\Sigma = -1$: balancing (negative) loop.

Assumptions and Conditions

1. The variable set V is finite.
2. Signs encode local monotonic influence only.
3. No quantitative magnitude is assumed.
4. Causal relations are interpreted structurally, not statistically.
5. The CLD does not specify functional form.

CLDs represent qualitative structure, not dynamical equations.

Proof or Mathematical Development

Suppose the system admits a differentiable representation

$$\dot{x}_i = f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n.$$

Define the Jacobian matrix

$$J(x) = \left(\frac{\partial f_i}{\partial x_j} \right).$$

Then a CLD corresponds to the sign structure of J:

$$\sigma_{ji} = \operatorname{sign}\left(\frac{\partial f_i}{\partial x_j} \right),$$

whenever this derivative is nonzero.

Thus, a CLD abstracts the sign pattern of the system Jacobian.

For a feedback loop L, the product

$$\Sigma = \prod_{(i,j) \in L} \operatorname{sign}\left(\frac{\partial f_j}{\partial x_i} \right)$$

captures whether perturbations are amplified or counteracted along that cycle.

In linear systems

$$\dot{x} = Ax,$$

the CLD corresponds to the sign structure of matrix A.

Thus, causal loop diagrams are qualitative projections of underlying dynamical systems.

Structural Role

Causal loop diagrams:

- Encode feedback topology.
- Identify reinforcing and balancing cycles.
- Provide a bridge between qualitative reasoning and formal dynamics.
- Serve as the precursor to adjacency matrices and graph-theoretic analysis.

They do not define trajectories. They define interaction structure.

Dynamics arise only after functional forms are specified.

Structural Compression

Invariant

A causal loop diagram is a signed directed graph:

$$\mathcal{G} = (V, E, \sigma).$$

Mechanism

Feedback loops correspond to directed cycles; loop sign is the product of edge signs.

Cross-System Echo

- Linear Algebra: sign structure of the Jacobian matrix.
- Network Theory: adjacency matrices and cycle detection.
- Control Theory: feedback interconnection topology.
- Economics: reinforcing vs stabilizing policy loops.

Stock-and-Flow Models

Position in Knowledge Graph

This node formalizes stock-and-flow structure as a quantitative refinement of causal loop diagrams (1.3).

It builds on: - Feedback mechanisms (1.1) - Causal interaction structure (1.3)

It prepares for: - Time delays and oscillations (1.5) - Formal dynamical systems (2.1)

Stock-and-flow models introduce conservation structure and accumulation.

Formal Statement / Definition / Construction

Let $S = \{s_1, \dots, s_n\}$ be a finite set of **stocks**.

A **stock** is a state variable

$$s_i : \mathbb{R}_{\geq 0} \to \mathbb{R},$$

interpreted as an accumulated quantity.

Let $F = \{f_1, \dots, f_m\}$ be a finite set of **flows**.

Each flow

$$f_k : \mathbb{R}^n \to \mathbb{R}$$

represents a rate of change.

A **stock-and-flow model** is defined by differential equations of the form

$$\dot{s}_i(t) = \sum_{k \in \mathcal{I}_i} f_k(s(t)) - \sum_{k \in \mathcal{O}_i} f_k(s(t)),$$

where:

- $\mathcal{I}_i \subseteq \{1, \dots, m\}$ indexes inflows to stock s_i,
- $\mathcal{O}_i \subseteq \{1, \dots, m\}$ indexes outflows from stock s_i,
- $s(t) = (s_1(t), \dots, s_n(t))$.

Equivalently, in vector form:

$$\dot{s}(t) = Bf(s(t)),$$

where:

- $f(s) \in \mathbb{R}^m$ is the flow vector,
- $B \in \mathbb{R}^{n \times m}$ is the incidence matrix, with entries in $\{-1, 0, +1\}$, encoding inflow $(+1)$ and outflow (-1) structure.

Assumptions and Conditions

1. Stocks are differentiable functions of time.
2. Flow functions are well-defined on $\mathbb{R}^n$.
3. If existence and uniqueness of solutions are required, assume flows are locally Lipschitz.
4. Conservation structure is encoded in matrix B.

No assumption of linearity is imposed.

Proof or Mathematical Development

A stock accumulates net inflow:

$$s_i(t) = s_i(0) + \int_0^t \dot{s}_i(\tau)\, d\tau.$$

Substituting the flow definition:

$$s_i(t) = s_i(0) + \int_0^t \left(\sum_{k \in \mathcal{I}_i} f_k(s(\tau)) - \sum_{k \in \mathcal{O}_i} f_k(s(\tau)) \right) d\tau.$$

Thus, stocks are integrators of flows.

If the system is closed (no external inflow or outflow), then total stock

$$S_{\text{total}}(t) = \sum_{i=1}^{n} s_i(t)$$

satisfies

$$\frac{d}{dt} S_{\text{total}}(t) = \sum_{i=1}^{n} \dot{s}_i(t).$$

Using matrix form:

$$\frac{d}{dt} S_{\text{total}}(t) = \mathbf{1}^\top B f(s(t)).$$

If

$$\mathbf{1}^\top B = 0,$$

then

$$\frac{d}{dt} S_{\text{total}}(t) = 0,$$

so total stock is conserved.

Thus conservation laws are linear algebraic constraints on B.

Structural Role

Stock-and-flow models introduce:

- Accumulation
- Conservation structure
- Integration over time
- Physical interpretability

They refine causal loop diagrams by distinguishing:

- Level variables (stocks)
- Rate variables (flows)

All continuous-time dynamical systems can be expressed in stock-and-flow form.

They are the bridge between qualitative feedback diagrams and quantitative differential equations.

Structural Compression

Invariant

Stocks evolve by integration of net flows:

$$\dot{s} = Bf(s).$$

Mechanism

Incidence matrix B encodes conservation and flow topology.

Cross-System Echo

- Linear Algebra: conservation corresponds to null space of $\mathbf{1}^\top B$.
- Physics: mass and energy balance equations.
- Economics: capital accumulation models.
- Biology: population dynamics with birth–death flows.

Time Delays and Oscillations

Position in Knowledge Graph

This node extends feedback systems by incorporating temporal delay.

It builds on: - Positive and negative feedback (1.2) - Stock-and-flow models (1.4)

It prepares for: - Phase portraits (2.2) - Limit cycles (2.5) - Stability analysis (3.1)

Time delay transforms simple feedback into oscillatory or unstable behavior.

Formal Statement / Definition / Construction

Let $x(t) \in \mathbb{R}$ be a stock variable.

A **delayed feedback system** is defined by a delay differential equation:

$$\dot{x}(t) = f\big(x(t), x(t-\tau)\big),$$

where:

- $\tau > 0$ is a fixed time delay,
- $f : \mathbb{R}^2 \to \mathbb{R}$.

A linear delayed negative feedback system takes the form:

$$\dot{x}(t) = -ax(t-\tau), \quad a > 0.$$

Assumptions and Conditions

1. $\tau > 0$ is constant.
2. The initial history function
$$x(t) = \phi(t), \quad t \in [-\tau, 0]$$
is continuous.
3. For existence and uniqueness, assume f is locally Lipschitz.
4. Linear stability analysis assumes exponential trial solutions.

No assumption of small delay is imposed.

Proof or Mathematical Development

Consider the linear system:

$$\dot{x}(t) = -ax(t-\tau).$$

Seek exponential solutions of the form:

$$x(t) = e^{\lambda t}.$$

Substitution yields:

$$\lambda e^{\lambda t} = -a e^{\lambda(t-\tau)}.$$

Canceling $e^{\lambda t} \neq 0$:

$$\lambda = -a e^{-\lambda\tau}.$$

This is the **characteristic equation**.

Rewriting:

$$\lambda e^{\lambda\tau} + a = 0.$$

Oscillatory solutions occur when $\lambda = i\omega$ with $\omega \neq 0$.

Substitute:

$$i\omega e^{i\omega\tau} + a = 0.$$

Using Euler's formula:

$$e^{i\omega\tau} = \cos(\omega\tau) + i\sin(\omega\tau).$$

Separate real and imaginary parts:

$$\begin{cases} -\omega\sin(\omega\tau) + a = 0, \\ \omega\cos(\omega\tau) = 0. \end{cases}$$

From the second equation:

$$\cos(\omega\tau) = 0 \quad \Rightarrow \quad \omega\tau = \frac{\pi}{2} + k\pi.$$

Substituting into the first equation yields:

$$\omega = a.$$

Thus oscillations occur when delay satisfies

$$a\tau = \frac{\pi}{2} + k\pi.$$

For sufficiently large τ, characteristic roots cross into the right half-plane, producing instability and sustained oscillations.

Therefore:

Negative feedback with delay can produce oscillatory dynamics.

Structural Role

Time delay introduces:

- Phase lag
- Memory
- Infinite-dimensional state structure

Without delay, linear negative feedback in one dimension cannot oscillate.

Delay creates the possibility of:

- Cycles
- Overshoot
- Instability
- Hopf-type transitions (formalized in later modules)

Delay converts stabilizing feedback into oscillatory dynamics.

Structural Compression

Invariant

Delayed feedback satisfies the transcendental characteristic equation:

$$\lambda e^{\lambda \tau} + a = 0.$$

Mechanism

Exponential trial solutions reveal that phase lag induces complex eigenvalues.

Cross-System Echo

- Control Theory: phase margin and oscillation.
- Economics: delayed expectations generate business cycles.
- Biology: predator–prey lag dynamics.
- Networks: signal propagation delays induce synchronization failure.