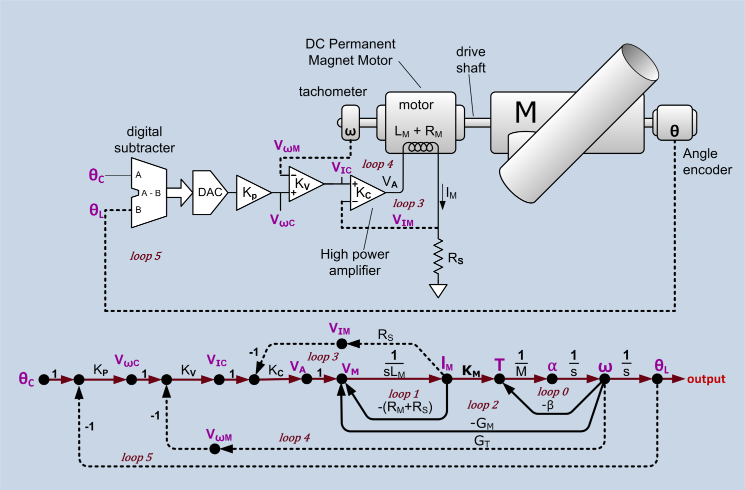
Network Theory I: Electrical Circuits and Signal-Flow Graphs

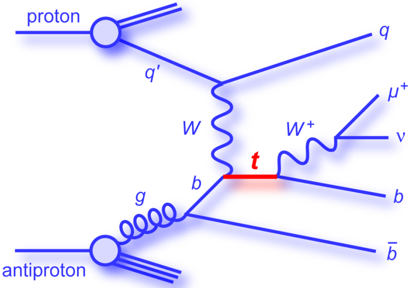
John Baez, Jason Erbele, Brendan Fong



The category with vector spaces as objects and linear maps as morphisms becomes symmetric monoidal with the usual $\otimes$.

The category with vector spaces as objects and linear maps as morphisms becomes symmetric monoidal with the usual $\otimes$.

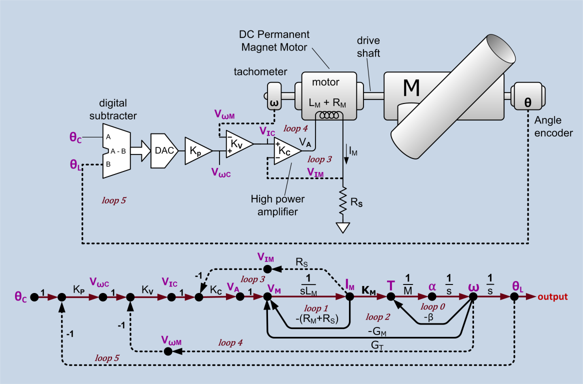
In quantum field theory, ‘Feynman diagrams’ are pictures of morphisms in this symmetric monoidal category:



But the category of vector spaces also becomes symmetric monoidal with direct sum, $\oplus$, as its 'tensor product'. Today we will explore this.

But the category of vector spaces also becomes symmetric monoidal with direct sum, $\oplus$, as its 'tensor product'. Today we will explore this.

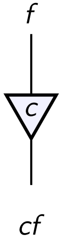
Control theorists use 'signal-flow graphs' to describe how signals flow through a system and interact:



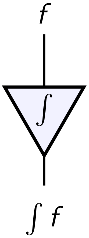
Think of a signal as a smooth real-valued function of time:

$$f : \mathbb{R} \to \mathbb{R}$$

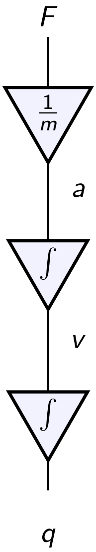
We can multiply a signal by a constant and get a new signal:



We can integrate a signal:



Here is what happens when you push on a mass m with a time-dependent force F:



Integration introduces an ambiguity: the constant of integration. But electrical engineers often use Laplace transforms to write signals as linear combinations of exponentials

$$f(t) = e^{-st} \quad \text{for some } s > 0$$

Then they define

$$(\textstyle\int f)(t) = \frac{e^{-st}}{s}$$

Integration introduces an ambiguity: the constant of integration. But electrical engineers often use Laplace transforms to write signals as linear combinations of exponentials

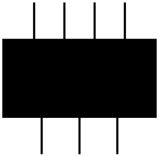
$$f(t) = e^{-st} \quad \text{for some } s > 0$$

Then they define

$$(\textstyle\int f)(t) = \frac{e^{-st}}{s}$$

This lets us think of integration as a special case of scalar multiplication! We extend our field of scalars from $\mathbb{R}$ to $\mathbb{R}(s)$, the field of rational real functions in one variable s.

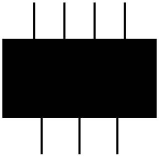
Let us be general and work with an arbitrary field k. For us, any signal-flow graph with m input edges and n output edges



will stand for a linear map

$$F\colon k^m \to k^n$$

Let us be general and work with an arbitrary field k. For us, any signal-flow graph with m input edges and n output edges



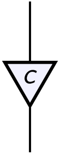
will stand for a linear map

$$F\colon k^m \to k^n$$

In other words: signal-flow graphs are pictures of morphisms in $\mathrm{FinVect}_k$, the category of finite-dimensional vector spaces over k... *where we make this into a monoidal category using* $\oplus$*, not* $\otimes$*.*

We build these pictures from a few simple 'generators'.

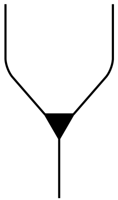
First, we have scalar multiplication:



This is a notation for the linear map

$$\begin{array}{ccc} k & \rightarrow & k \\ f & \mapsto & cf \end{array}$$

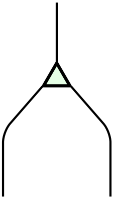
Second, we can add two signals:



This is a notation for

$$+\colon k \oplus k \to k$$

Third, we can 'duplicate' a signal:



This is a notation for the diagonal map

$$\begin{array}{rccc} \Delta\colon & k & \to & k \oplus k \\ & f & \mapsto & (f, f) \end{array}$$

Fourth, we can 'delete' a signal:

This is a notation for the linear map

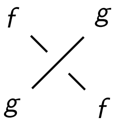
$$\begin{array}{rcl} k & \to & \{0\} \\ f & \mapsto & 0 \end{array}$$

Fifth, we have the zero signal:

This is a notation for the linear map

$$\begin{array}{ccc} \{0\} & \to & k \\ 0 & \mapsto & 0 \end{array}$$

Furthermore, ($\mathrm{FinVect}_k$, $\oplus$) is a *symmetric* monoidal category. This means we have a 'braiding': a way to switch two signals:

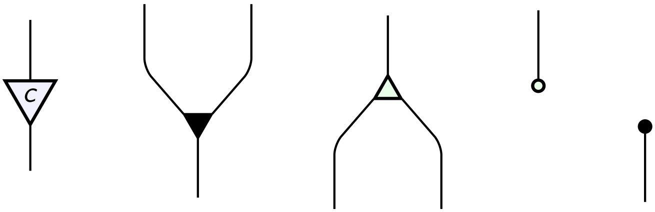


This is a notation for the linear map

$$\begin{array}{ccc} k \oplus k & \rightarrow & k \oplus k \\ (f, g) & \mapsto & (g, f) \end{array}$$

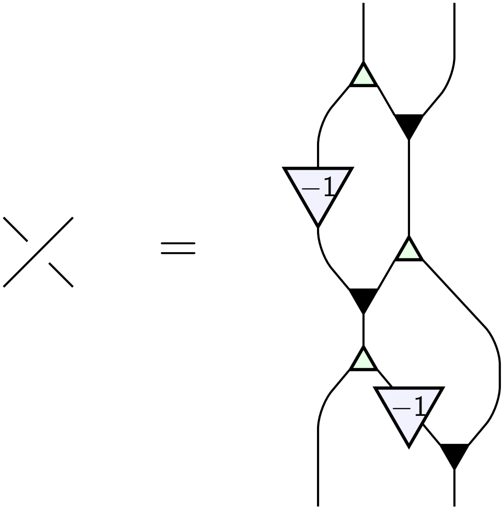
In a symmetric monoidal category, the braiding must obey a few axioms. I won't list them here, since they are easy to find.

From these ‘generators’:



together with the braiding, we can build complicated signal-flow graphs. In fact, we can describe *any* linear map $F\colon k^m \to k^n$ this way!

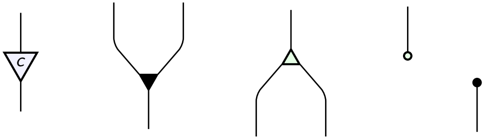
But these generators obey some unexpected relations:



Luckily, we can derive *all* the relations from some very nice ones!

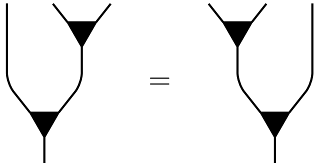
Theorem (Jason Erbele)

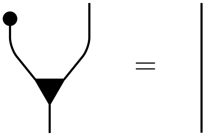
$FinVect_k$ is equivalent to the symmetric monoidal category generated by the object k and these morphisms:

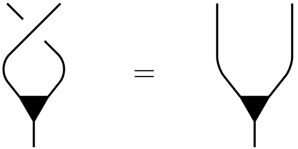


where $c \in k$, with the following relations.

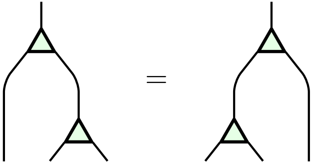
Addition and zero make k into a commutative monoid:

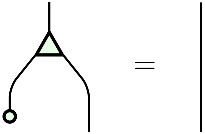


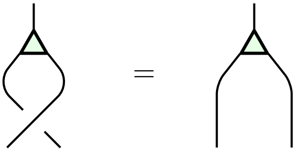




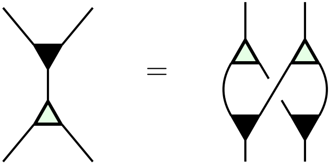
Duplication and deletion make k into a cocommutative comonoid:

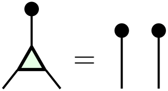


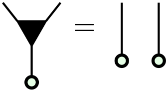


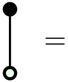


The monoid and comonoid operations are compatible, as in a bialgebra:

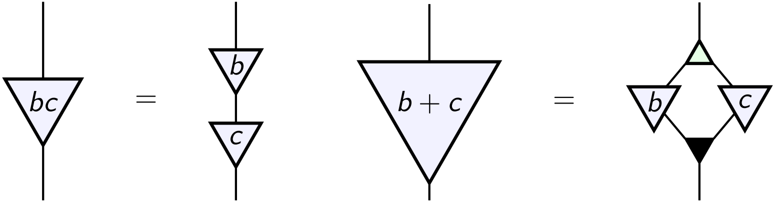


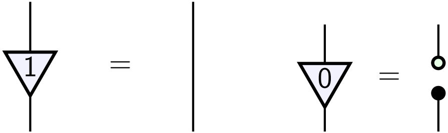




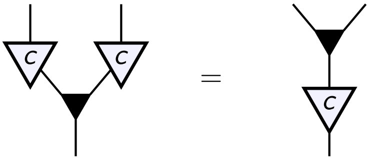


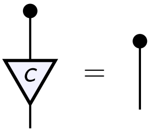
The ring structure of k can be recovered from the generators:



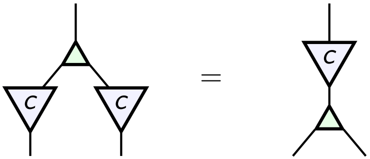


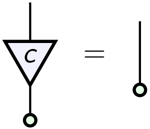
Scalar multiplication is linear (compatible with addition and zero):





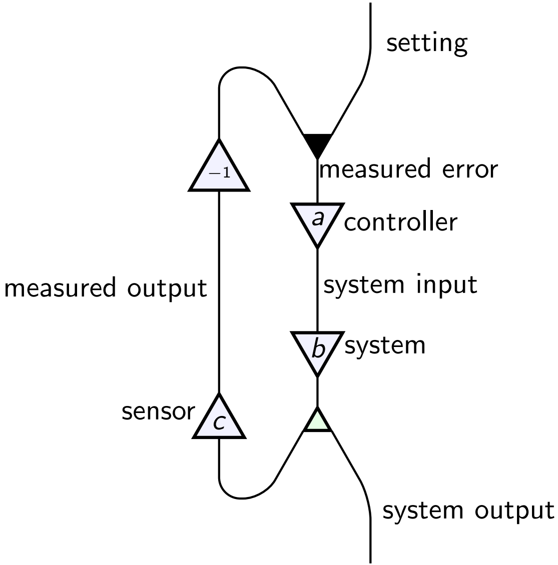
Scalar multiplication is 'colinear' (compatible with duplication and deletion):





Those are all the relations we need!

However, control theory also needs more general signal-flow graphs, which have 'feedback loops':



Feedback is the most important concept in control theory: letting the output of a system affect its input. For this we should let wires 'bend back':

These aren't linear functions — they're linear relations!

A **linear relation** $F\colon U \rightsquigarrow V$ from a vector space U to a vector space V is a linear subspace $F \subseteq U \oplus V$.

A **linear relation** $F\colon U \rightsquigarrow V$ from a vector space U to a vector space V is a linear subspace $F \subseteq U \oplus V$.

We can compose linear relations $F\colon U \rightsquigarrow V$ and $G\colon V \rightsquigarrow W$ and get a linear relation $G \circ F\colon U \rightsquigarrow W$:

$$G \circ F = \{(u, w)\colon \ \exists v \in V \quad (u, v) \in F \text{ and } (v, w) \in G\}.$$

A linear map $\phi\colon U \to V$ gives a linear relation $F\colon U \rightsquigarrow V$, namely the graph of that map:

$$F = \{(u, \phi(u)) : u \in U\}$$

Composing linear maps becomes a special case of composing linear relations.

A linear map $\phi\colon U \to V$ gives a linear relation $F\colon U \rightsquigarrow V$, namely the graph of that map:

$$F = \{(u, \phi(u)) : u \in U\}$$

Composing linear maps becomes a special case of composing linear relations.

There is a symmetric monoidal category FinRel_k with finite-dimensional vector spaces over the field k as objects and linear relations as morphisms. This has $\mathrm{FinVect}_k$ as a subcategory.

A linear map $\phi\colon U \to V$ gives a linear relation $F\colon U \rightsquigarrow V$, namely the graph of that map:

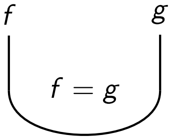
$$F = \{(u, \phi(u)) : u \in U\}$$

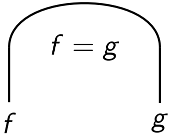
Composing linear maps becomes a special case of composing linear relations.

There is a symmetric monoidal category FinRel_k with finite-dimensional vector spaces over the field k as objects and linear relations as morphisms. This has $\mathrm{FinVect}_k$ as a subcategory.

Fully general signal-flow graphs are pictures of morphisms in FinRel_k, typically with $k = \mathbb{R}(s)$.

Jason Erbele showed that besides the previous generators of $\mathrm{FinVect}_k$, we only need two more morphisms to generate all the morphisms in FinRel_k: the 'cup' and 'cap'.





These linear relations say that when a signal goes around a bend in a wire, the signal coming out equals the signal going in!

More formally, the cup is the linear relation

$$\cup \colon k \oplus k \rightsquigarrow \{0\}$$

given by:

$$\cup \;=\; \{(f, f, 0) : f \in k\} \;\subseteq\; k \oplus k \oplus \{0\}$$

More formally, the cup is the linear relation

$$\cup \colon k \oplus k \rightsquigarrow \{0\}$$

given by:

$$\cup \;=\; \{(f, f, 0) : f \in k\} \;\subseteq\; k \oplus k \oplus \{0\}$$

Similarly, the cap is the linear relation

$$\cap \colon \{0\} \rightsquigarrow k \oplus k$$

given by:

$$\cap \;=\; \{(0, f, f) : f \in k\} \;\subseteq\; \{0\} \oplus k \oplus k$$

More formally, the cup is the linear relation

$$\cup \colon k \oplus k \rightsquigarrow \{0\}$$

given by:

$$\cup \;=\; \{(f, f, 0) : f \in k\} \;\subseteq\; k \oplus k \oplus \{0\}$$

Similarly, the cap is the linear relation

$$\cap \colon \{0\} \rightsquigarrow k \oplus k$$

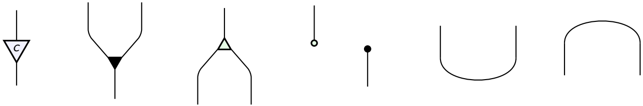
given by:

$$\cap \;=\; \{(0, f, f) : f \in k\} \;\subseteq\; \{0\} \oplus k \oplus k$$

These make (FinRel_k, $\oplus$) into a 'dagger-compact category'.

Theorem (Jason Erbele)

FinRel_k is equivalent to the symmetric monoidal category generated by the object k and these morphisms:



where $c \in k$, and an explicit list of relations.

Instead of listing the relations, let me just sketch what comes next!

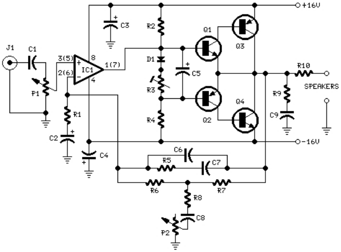
Instead of listing the relations, let me just sketch what comes next!

I have only talked about *linear* control theory. There is also a nonlinear version.

Instead of listing the relations, let me just sketch what comes next!

I have only talked about *linear* control theory. There is also a nonlinear version.

In both versions there's a general issue: engineers want to build devices that actually *implement* a given signal-flow graph. One way is to use electrical circuits. These are described using 'circuit diagrams':



In the linear case, there is bicategory Circ whose morphisms are circuit diagrams made of resistors, capacitors and inductors.

In the linear case, there is bicategory Circ whose morphisms are circuit diagrams made of resistors, capacitors and inductors.

Thanks to work in progress by Brendan Fong, we know there is a *functor* from this bicategory to FinRel_k:

$$\mathrm{Z} \colon \mathrm{Circ} \to \mathrm{FinRel}_k$$

where $k = \mathbb{R}(s)$.

In the linear case, there is bicategory Circ whose morphisms are circuit diagrams made of resistors, capacitors and inductors.

Thanks to work in progress by Brendan Fong, we know there is a *functor* from this bicategory to FinRel_k:

$$Z\colon \mathrm{Circ} \to \mathrm{FinRel}_k$$

where $k = \mathbb{R}(s)$.

This functor says, for any circuit diagram, how the voltages and currents on the input wires are related to those on the ouput wires.

However, we do not get *arbitrary* linear relations this way. The space of voltages and currents on n wires:

$$k^n \oplus k^n$$

is a **symplectic** vector space, meaning that it's equipped with a skew-symmetric nondegenerate bilinear form:

$$\omega((V_1, I_1), (V_2, I_2)) = I_1 \cdot V_2 - I_2 \cdot V_1$$

called the **symplectic 2-form**.

This is similar to how in classical mechanics, the space of positions and momenta of a collection of particles is a symplectic 2-form on $\mathbb{R}^n \oplus \mathbb{R}^n$.

The linear relation

$$F \colon k^m \oplus k^m \rightsquigarrow k^n \oplus k^n$$

we get from a linear circuit is always a **Lagrangian relation**, meaning that

$$F \subseteq (k^m \oplus k^m) \oplus (k^n \oplus k^n)$$

is a **Lagrangian subspace**: a maximal subspace on which the symplectic 2-form vanishes.

Similarly, in classical mechanics, the inital and final positions/ momenta of a collection of particles lie in a Lagrangian submanifold of $\mathbb{R}^n \oplus \mathbb{R}^n$.

So, we can see the beginnings of an interesting relation between:

- control theory
- electrical engineering
- category theory
- symplectic geometry

So, we can see the beginnings of an interesting relation between:

- control theory
- electrical engineering
- category theory
- symplectic geometry

This should become even more interesting when we study nonlinear systems. And as we move from the networks important in *human-engineered* systems to those important in *biology* and *ecology*, the mathematics should become even more rich!