Categories in Control

John C. Baez
Department of Mathematics
University of California
Riverside CA 92521
USA
Centre for Quantum Technologies
National University of Singapore
Singapore 117543
baez@math.ucr.edu

Jason Erbele
Department of Mathematics
University of California
Riverside CA 92521
USA
erbele@math.ucr.edu

Control theory uses 'signal-flow diagrams' to describe processes where real-valued functions of time are added, multiplied by scalars, differentiated and integrated, duplicated and deleted. These diagrams can be seen as string diagrams for the symmetric monoidal category $\mathrm{FinVect}_k$ of finite-dimensional vector spaces over the field of rational functions $k = \mathbb{R}(s)$, where the variable s acts as differentiation and the monoidal structure is direct sum rather than the usual tensor product of vector spaces. For any field k we give a presentation of $\mathrm{FinVect}_k$ in terms of the generators used in signal-flow diagrams. A broader class of signal-flow diagrams also includes 'caps' and 'cups' to model feedback. We show these diagrams can be seen as string diagrams for the symmetric monoidal category FinRel_k, where objects are still finite-dimensional vector spaces but the morphisms are linear relations. We also give a presentation for FinRel_k. The relations say, among other things, that the 1-dimensional vector space k has two special commutative †-Frobenius structures, such that the multiplication and unit of either one and the comultiplication and counit of the other fit together to form a bimonoid. This sort of structure, but with tensor product replacing direct sum, is familiar from the 'ZX-calculus' obeyed by a finite-dimensional Hilbert space with two mutually unbiased bases.

1 Introduction

Control theory is the branch of engineering that focuses on manipulating 'open systems'—systems with inputs and outputs—to achieve desired goals. In control theory, 'signal-flow diagrams' are used to describe linear ways of manipulating signals, which we will take here to be smooth real-valued functions of time [11]. For a category theorist, at least, it is natural to treat signal-flow diagrams as string diagrams in a symmetric monoidal category [12, 13]. This forces some small changes of perspective, which we discuss below, but more important is the question: *which symmetric monoidal category?*

In [3] we argue that the answer is: the category FinRel_k of finite-dimensional vector spaces over a certain field k, but with *linear relations* rather than linear maps as morphisms, and *direct sum* rather than tensor product providing the symmetric monoidal structure. This current paper is a summary of those results. We use the field $k = \mathbb{R}(s)$ consisting of rational functions in one real variable s. This variable has the meaning of differentation. A linear relation from k^m to k^n is thus a system of linear constant-coefficient ordinary differential equations relating m 'input' signals and n 'output' signals.

We provide a complete 'generators and relations' picture of this symmetric monoidal category, with the generators being familiar components of signal-flow diagrams. Filippo Bonchi, Paweł Sobociński and Fabio Zanasi [5] independently gave an equivalent presentation using a different approach of composing PROPs. The answer is connected to ideas familiar in the diagrammatic approach to quantum theory in a beautiful way. Quantum theory also involves linear algebra, but uses linear maps between Hilbert spaces as morphisms, and the tensor product of Hilbert spaces provides the symmetric monoidal structure.

Submitted to:
QPL 2015

2 Signal-flow diagrams

There are several basic operations that one wants to perform when manipulating signals. The simplest is multiplying a signal by a scalar. A signal can be amplified by a constant factor:

$$f \mapsto cf$$

where $c \in \mathbb{R}$. We can write this as a string diagram as in Figure 1. The labels f and cf on top and bottom are just for explanatory purposes and not really part of the diagram. Control theorists often draw arrows on the wires to indicate which carries the input f and which the output cf, but this goes against the convention for arrows on wires of string diagrams to distinguish objects from their duals. Fortunately we can avoid confusion by dropping all the arrows: we will obtain a compact closed category where each object is its own dual, and the triangular shape of multiplication distinguishes f from cf.

A signal can also be integrated with respect to the time variable:

$$f \mapsto \int f.$$

Mathematicians typically take differentiation as fundamental, but engineers sometimes prefer integration, because it is more robust against small perturbations. In the end it will not matter much here. We again draw integration as a string diagram in Figure 1. Since this looks like the diagram for scalar

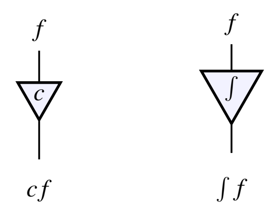


Figure 1: Scalar multiplication and integration

multiplication, it is natural to extend $\mathbb{R}$ to $\mathbb{R}(s)$, the field of rational functions of a variable s which stands for differentiation. Then differentiation becomes a special case of scalar multiplication, namely multiplication by s, and integration becomes multiplication by $1/s$. Engineers accomplish the same effect with Laplace transforms, since differentiating a signal f is equivalent to multiplying its Laplace transform

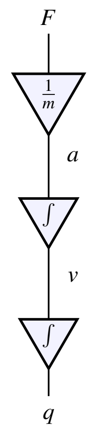
$$(\mathscr{L}f)(s) = \int_0^\infty f(t)e^{-st}\,dt$$

by the variable s. Another option is to use the Fourier transform: differentiating f is equivalent to multiplying its Fourier transform

$$(\mathscr{F}f)(\omega) = \int_{-\infty}^\infty f(t)e^{-i\omega t}\,dt$$

by $i\omega$. Of course, the function f needs to be sufficiently well-behaved to justify calculations involving its Laplace or Fourier transform. At a more basic level, it also requires some work to treat integration as the two-sided inverse of differentiation. Engineers do this by considering signals that vanish for $t < 0$, and choosing the antiderivative that vanishes under the same condition. Luckily all these issues can be side-stepped in a formal treatment of signal-flow diagrams: we can simply treat signals as living in an unspecified vector space over the field $\mathbb{R}(s)$, or more generally over an arbitrary field k.

The simplest possible signal processor is a rock, which takes the 'input' given by the force F on the rock and produces as 'output' the rock's position q. Thanks to Newton's second law $F = ma$, we can describe this using a signal-flow diagram:

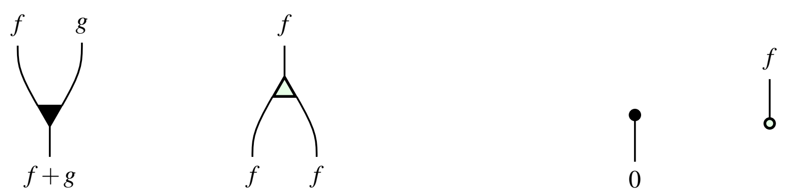


Here composition of morphisms is drawn in the usual way, by attaching the output wire of one morphism to the input wire of the next.

To build more interesting machines we need more building blocks, such as addition and duplication:

$$+\colon (f,g) \mapsto f+g$$
$$\Delta\colon f \mapsto (f,f).$$

When these linear maps are written as matrices, their matrices are transposes of each other. This is reflected in the string diagrams for addition and duplication, as seen in Figure 2a. The second is essentially



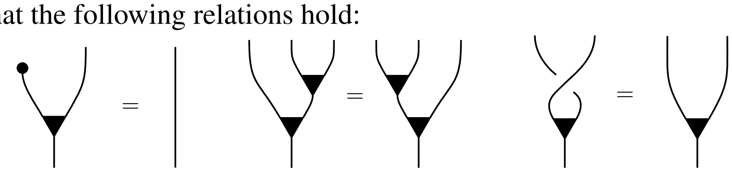
(a) Addition and duplication

(b) Zero and deletion

Figure 2

an upside-down version of the first. However, we draw addition as a dark triangle and duplication as a light one because we will later want another way to 'turn addition upside-down' that does *not* give duplication. As an added bonus, a light upside-down triangle resembles the Greek letter Δ, the usual symbol for duplication.

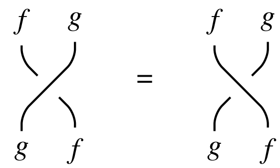
While they are typically not considered worthy of mention in control theory, for completeness we must include two other building blocks. One is the zero map from $\{0\}$ to our field k, which we denote as 0 and draw with a black dot. The other is the zero map from k to $\{0\}$, sometimes called 'deletion', which we denote as ! and draw as a light circle, as in Figure 2b. Addition and zero make k into a commutative monoid, meaning that the following relations hold:



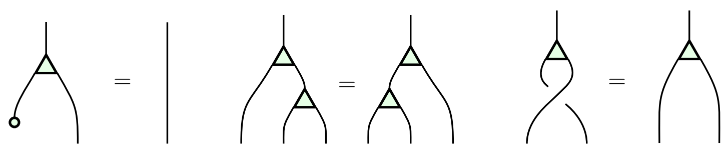
The equation at right is the commutative law, and the crossing of strands is the ‘braiding’

$$B\colon (f,g) \mapsto (g,f)$$

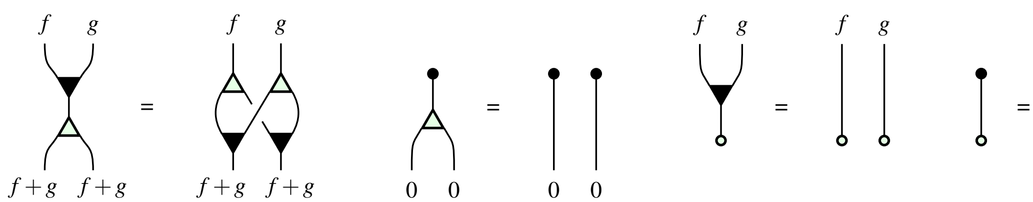
by which we switch two signals. In fact this braiding is a ‘symmetry’, so it does not matter which strand goes over which:



Dually, duplication and deletion make k into a cocommutative comonoid. This means that if we reflect the equations obeyed by addition and zero across the horizontal axis and turn dark operations into light ones, we obtain another set of valid equations:



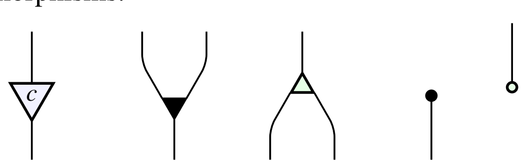
There are also relations between the monoid and comonoid operations. For example, adding two signals then duplicating the result gives the same output as duplicating each signal and then adding the results.



These diagrams are familiar in the theory of Hopf algebras, or more generally bialgebras. Here they demonstrate that the monoid operations on k are comonoid homomorphisms—or equivalently, that the comonoid operations are monoid homomorphisms. We summarize this by saying that k is a **bimonoid**.

So far all our string diagrams denote linear maps. We can treat these as morphisms in the category $\mathrm{FinVect}_k$, where objects are finite-dimensional vector spaces over a field k and morphisms are linear maps. This category is equivalent to a skeleton where the only objects are vector spaces k^n for $n \geq 0$, and then morphisms can be seen as $n \times m$ matrices. The space of signals is a vector space V over k which may not be finite-dimensional, but this does not cause a problem: an $n \times m$ matrix with entries in k still defines a linear map from V^n to V^m in a functorial way.

In applications of string diagrams to quantum theory [4, 9], we make $\mathrm{FinVect}_k$ into a symmetric monoidal category using the tensor product of vector spaces. In control theory, we instead make $\mathrm{FinVect}_k$ into a symmetric monoidal category using the *direct sum* of vector spaces. In Lemma 1 of [3] we prove that for any field k, $\mathrm{FinVect}_k$ with direct sum is generated as a symmetric monoidal category by the one object k together with these morphisms:



where $c \in k$ is arbitrary.

However, these generating morphisms obey some unexpected relations! For example, we have:

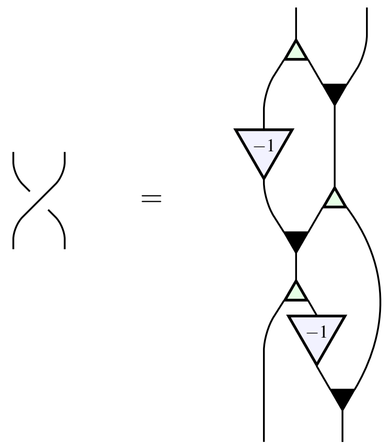
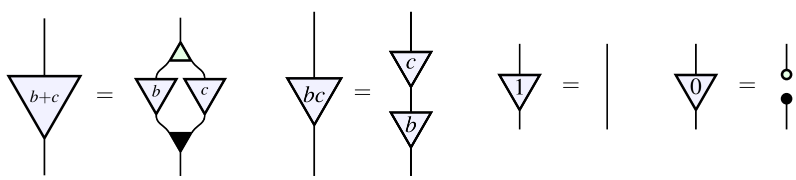


Figure 3

Thus, it is important to find a complete set of relations obeyed by these generating morphisms, thus obtaining a presentation of $\mathrm{FinVect}_k$ as a symmetric monoidal category. We do this in Theorem 2 of [3]. In brief, these relations say:

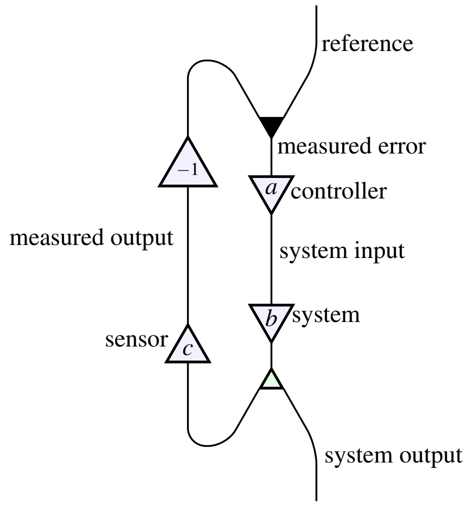
1. $(k, +, 0, \Delta, !)$ is a bicommutative bimonoid;
2. the rig operations of k can be recovered from the generating morphisms;
3. all the generating morphisms commute with scalar multiplication.

Item (2) means that $+$, $\cdot$, 0 and 1 in the field k can be expressed in terms of signal-flow diagrams as:



Multiplicative inverses cannot be so expressed, so our signal-flow diagrams so far do not know that k is a field. Additive inverses also cannot be expressed in this way. So, we expect that a version of Theorem 2 will hold whenever k is a mere rig: that is, a 'ring without negatives', like $\mathbb{N}$. The one change is that instead of working with vector spaces, we should work with finitely presented free k-modules. Since Figure 3 gives a way to write the braiding using other generating morphisms, we should expect $\mathrm{FinVect}_k$ can be presented as merely a monoidal category without explicitly including the braiding. Lafont [16] proved this for the case where k is the field with two elements and stated the result for a general field.

While we have made a step towards understanding the category-theoretic underpinnings of control theory with Theorem 2, it does not treat signal-flow diagrams that include 'feedback'. Feedback is one of the most fundamental concepts in control theory because a control system without feedback may be highly sensitive to disturbances or unmodeled behavior. Feedback allows these uncontrolled behaviors to be mollified. As a string diagram, a basic feedback system might look schematically like this:



The user inputs a ‘reference’ signal, which is fed into a controller, whose output is fed into a system, or ‘plant’, which in turn produces its own output. But then the system’s output is duplicated, and one copy is fed into a sensor, whose output is subtracted (or if we prefer, added) from the reference signal.

In string diagrams—unlike in the usual thinking on control theory—it is essential to be able to read any diagram from top to bottom as a composite of tensor products of generating morphisms. Thus, to incorporate the idea of feedback, we need two more generating morphisms. These are the ‘cup’ and ‘cap’. These are not maps: they are relations. The cup imposes the relation that its two inputs be equal,

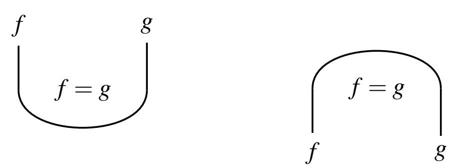


Figure 4: cup and cap

while the cap does the same for its two outputs. This describes the way signals flow around a bend in a wire.

To make this precise, we use a category called FinRel_k. An object of this category is a finite-dimensional vector space over k, while a morphism from U to V, denoted $L\colon U \nrightarrow V$, is a **linear relation**, meaning a linear subspace

$$L \subseteq U \oplus V.$$

In particular, when $k = \mathbb{R}(s)$, a linear relation $L\colon k^m \nrightarrow k^n$ is just an arbitrary system of constant-coefficient linear ordinary differential equations relating m input variables and n output variables.

Since the direct sum $U \oplus V$ is also the cartesian product of U and V, a linear relation is indeed a relation in the usual sense, but with the property that if $u \in U$ is related to $v \in V$ and $u' \in U$ is related to $v' \in V$ then $cu + c'u'$ is related to $cv + c'v'$ whenever $c, c' \in k$. We compose linear relations $L\colon U \nrightarrow V$ and $L'\colon V \nrightarrow W$ as follows:

$$L'L = \{(u,w)\colon\ \exists\, v \in V \ \ (u,v) \in L \text{ and } (v,w) \in L'\}.$$

Any linear map $f\colon U \to V$ gives a linear relation $F\colon U \nrightarrow V$, namely the graph of that map:

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

Composing linear maps thus becomes a special case of composing linear relations, so $\mathrm{FinVect}_k$ becomes a subcategory of FinRel_k. Furthermore, we can make FinRel_k into a monoidal category using direct sums, and it becomes symmetric monoidal using the braiding already present in $\mathrm{FinVect}_k$.

In these terms, the **cup** is the linear relation

$$\cup\colon k^2 \nrightarrow \{0\}$$

given by

$$\cup \;=\; \{(x,x,0) : x \in k\} \;\subseteq\; k^2 \oplus \{0\},$$

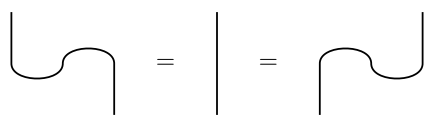
while the **cap** is the linear relation

$$\cap\colon \{0\} \nrightarrow k^2$$

given by

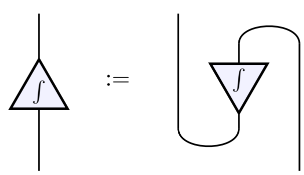
$$\cap \;=\; \{(0,x,x) : x \in k\} \;\subseteq\; \{0\} \oplus k^2.$$

These obey the **zigzag relations**:



Thus, they make FinRel_k into a compact closed category where k, and thus every object, is its own dual.

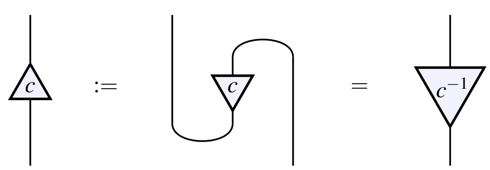
Besides feedback, one of the things that make the cap and cup useful is that they allow any morphism $L\colon U \nrightarrow V$ to be 'plugged in backwards' and thus 'turned around'. For instance, turning around integration, we obtain differentiation, denoted as 'integration pointing up':



In general, using caps and cups we can turn around any linear relation $L\colon U \nrightarrow V$ and obtain a linear relation $L^\dagger\colon V \nrightarrow U$, called the **adjoint** of L, which turns out to given by

$$L^\dagger = \{(v,u) : (u,v) \in L\}.$$

For example, if $c \in k$ is nonzero, the adjoint of scalar multiplication by c is multiplication by c^{-1}:

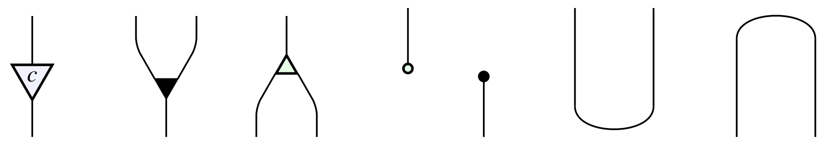


Thus, caps and cups allow us to express multiplicative inverses in terms of signal-flow diagrams! One might think that a problem arises when when $c = 0$, but no: the adjoint of scalar multiplication by 0 is

$$\{(0,x) : x \in k\} \subseteq k \oplus k,$$

which is not a linear map anymore, but is still a linear relation. In fact, when k is replaced by a ring, the adjoint of scalar multiplication by c is just a linear relation whenever c is a zero-divisor.

In Lemma 3 of [3] we show that FinRel_k is generated, as a symmetric monoidal category, by these morphisms:



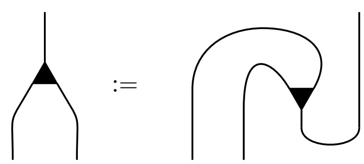
where $c \in k$ is arbitrary.

We follow that with Theorem 4, which provides a complete set of relations obeyed by these generating morphisms, thus giving a presentation of FinRel_k as a symmetric monoidal category. To describe these relations, it is useful to work with adjoints of the generating morphisms. We have already seen that the adjoint of scalar multiplication by c is scalar multiplication by c^{-1}, except when $c = 0$. Taking adjoints of the other four generating morphisms of $\mathrm{FinVect}_k$, we obtain four important but perhaps unfamiliar linear relations. We draw these as 'turned around' versions of the original generating morphisms:

- **Coaddition** is a linear relation from k to k^2 that holds when the two outputs sum to the input:

$$+^\dagger : k \nrightarrow k^2$$

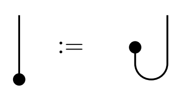
$$+^\dagger = \{(x,y,z) : \; x = y + z\} \subseteq k \oplus k^2$$



- **Cozero** is a linear relation from k to $\{0\}$ that holds when the input is zero:

$$0^\dagger : k \nrightarrow \{0\}$$

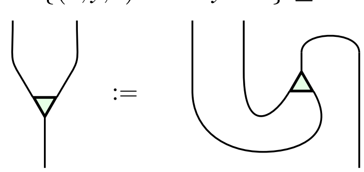
$$0^\dagger = \{(0,0)\} \subseteq k \oplus \{0\}$$



- **Coduplication** is a linear relation from k^2 to k that holds when the two inputs both equal the output:

$$\Delta^\dagger : k^2 \nrightarrow k$$

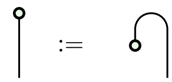
$$\Delta^\dagger = \{(x,y,z) : \; x = y = z\} \subseteq k^2 \oplus k$$



- **Codeletion** is a linear relation from $\{0\}$ to k that holds always:

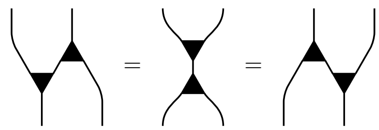
$$!^\dagger : \{0\} \nrightarrow k$$

$$!^\dagger = \{(0,x)\} \subseteq \{0\} \oplus k$$



Since $+^\dagger, 0^\dagger, \Delta^\dagger$ and $!^\dagger$ automatically obey turned-around versions of the relations obeyed by $+, 0, \Delta$ and $!$, we see that k acquires a *second* bicommutative bimonoid structure when considered as an object in FinRel_k.

Moreover, the four dark operations make k into a **Frobenius monoid**. This means that $(k, +, 0)$ is a monoid, $(k, +^\dagger, 0^\dagger)$ is a comonoid, and the **Frobenius relation** holds:

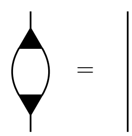


All three expressions in this equation are linear relations saying that the sum of the two inputs equal the sum of the two outputs.

The operation sending each linear relation to its adjoint extends to a contravariant functor

$$\dagger \colon \mathrm{FinRel}_k \to \mathrm{FinRel}_k,$$

which obeys a list of properties that are summarized by saying that FinRel_k is a '$\dagger$-compact' category [1, 18]. Because two of the operations in the Frobenius monoid $(k, +, 0, +^\dagger, 0^\dagger)$ are adjoints of the other two, it is a **$\dagger$-Frobenius monoid**. This Frobenius monoid is also **special**, meaning that comultiplication (in this case $+^\dagger$) followed by multiplication (in this case $+$) equals the identity:



It is also commutative—and cocommutative, but for Frobenius monoids this follows from commutativity.

Starting around 2008, commutative special $\dagger$-Frobenius monoids have become important in the categorical foundations of quantum theory, where they can be understood as 'classical structures' for quantum systems [10, 19]. The category FinHilb of finite-dimensional Hilbert spaces and linear maps is a $\dagger$-compact category, where any linear map $f \colon H \to K$ has an adjoint $f^\dagger \colon K \to H$ given by

$$\langle f^\dagger \phi, \psi \rangle = \langle \phi, f \psi \rangle$$

for all $\psi \in H, \phi \in K$. A commutative special $\dagger$-Frobenius monoid in FinHilb is then the same as a Hilbert space with a chosen orthonormal basis. The reason is that given an orthonormal basis ψ_i for a finite-dimensional Hilbert space H, we can make H into a commutative special $\dagger$-Frobenius monoid with multiplication $m \colon H \otimes H \to H$ given by

$$m(\psi_i \otimes \psi_j) = \begin{cases} \psi_i & i = j \\ 0 & i \neq j \end{cases}$$

and unit $i \colon \mathbb{C} \to H$ given by

$$i(1) = \sum_i \psi_i .$$

The comultiplication $m^\dagger$ duplicates basis states:

$$m^\dagger(\psi_i) = \psi_i \otimes \psi_i .$$

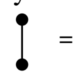
Conversely, any commutative special †-Frobenius monoid in FinHilb arises this way.

Considerably earlier, around 1995, commutative Frobenius monoids were recognized as important in topological quantum field theory. The reason, ultimately, is that the free symmetric monoidal category on a commutative Frobenius monoid is 2Cob, the category with 2-dimensional oriented cobordisms as morphisms: see Kock's textbook [14] and the many references therein. But the free symmetric monoidal category on a commutative special Frobenius monoid was worked out long before the rise of topological quantum field theory [7, 15, 17]: it is the category with finite sets as objects, where a morphism $f\colon X \to Y$ is an isomorphism class of cospans

$$X \longrightarrow S \longleftarrow Y.$$

This category can be made into a †-compact category in an obvious way, and then the 1-element set becomes a commutative special †-Frobenius monoid.

For all these reasons, it is interesting to find a commutative special †-Frobenius monoid lurking at the heart of control theory! However, the Frobenius monoid here has yet another property, which is more unusual. Namely, the unit $0\colon \{0\} \nrightarrow k$ followed by the counit $0^\dagger\colon k \nrightarrow \{0\}$ is the identity:



We call a special Frobenius monoid that also obeys this extra law **extra-special**. One can check that the free symmetric monoidal category on a commutative extra-special Frobenius monoid is the category with finite sets as objects, where a morphism $f\colon X \to Y$ is an equivalence relation on the disjoint union $X \sqcup Y$, and we compose $f\colon X \to Y$ and $g\colon Y \to Z$ by letting f and g generate an equivalence relation on $X \sqcup Y \sqcup Z$ and then restricting this to $X \sqcup Z$.

As if this were not enough, the light operations share many properties with the dark ones. In particular, these operations make k into a commutative extra-special †-Frobenius monoid in a second way. In summary:

- $(k,+,0,\Delta,!)$ is a bicommutative bimonoid;
- $(k,\Delta^\dagger,!^\dagger,+^\dagger,0^\dagger)$ is a bicommutative bimonoid;
- $(k,+,0,+^\dagger,0^\dagger)$ is a commutative extra-special †-Frobenius monoid;
- $(k,\Delta^\dagger,!^\dagger,\Delta,!)$ is a commutative extra-special †-Frobenius monoid.

It should be no surprise that with all these structures built in, signal-flow diagrams are a powerful method of designing processes.

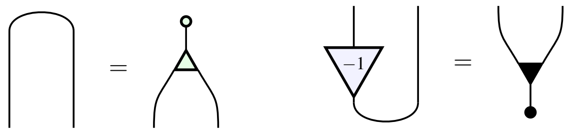
However, it is surprising that most of these structures are present in a seemingly very different context: the so-called 'ZX calculus', a diagrammatic formalism for working with complementary observables in quantum theory [8]. This arises naturally when one has an n-dimensional Hilbert space H with two orthonormal bases ψ_i, ϕ_i that are 'mutually unbiased', meaning that for all $1 \le i, j \le n$,

$$|\langle \psi_i, \phi_j \rangle|^2 = \frac{1}{n}.$$

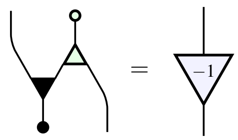
Each orthonormal basis makes H into commutative special †-Frobenius monoid in FinHilb. Moreover, the multiplication and unit of either one of these Frobenius monoids fits together with the comultiplication and counit of the other to form a bicommutative bimonoid. So, we have all the structure present in the list above—except that these Frobenius monoids are only extra-special if H is 1-dimensional.

The field k is also a 1-dimensional vector space, but this is a red herring: in FinRel_k *every* finite-dimensional vector space naturally acquires all four structures listed above, since addition, zero, duplication and deletion are well-defined and obey all the relations we have discussed. We focus on k in this paper simply because it generates all the objects FinRel_k via direct sum.

Finally, in FinRel_k the cap and cup are related to the light and dark operations as follows:



Note the curious factor of -1 in the second equation, which breaks some of the symmetry we have seen so far. This equation says that two elements $x, y \in k$ sum to zero if and only if $-x = y$. Using the zigzag relations, the two equations above give



We thus see that in FinRel_k, both additive and multiplicative inverses can be expressed in terms of the generating morphisms used in signal-flow diagrams.

In Theorem 4 of [3] we give a presentation of FinRel_k based on the ideas just discussed. In brief, FinRel_k is equivalent to the symmetric monoidal category generated by an object k and these morphisms:

1. addition $+\colon k^2 \nrightarrow k$
2. zero $0\colon \{0\} \nrightarrow k$
3. duplication $\Delta\colon k \nrightarrow k^2$
4. deletion $!\colon k \nrightarrow 0$
5. scalar multiplication $c\colon k \nrightarrow k$ for any $c \in k$
6. cup $\cup\colon k^2 \nrightarrow \{0\}$
7. cap $\cap\colon \{0\} \nrightarrow k^2$

obeying these relations:

1. $(k, +, 0, \Delta, !)$ is a bicommutative bimonoid;
2. $\cap$ and $\cup$ obey the zigzag equations;
3. $(k, +, 0, +^\dagger, 0^\dagger)$ is a commutative extra-special $\dagger$-Frobenius monoid;
4. $(k, \Delta^\dagger, !^\dagger, \Delta, !)$ is a commutative extra-special $\dagger$-Frobenius monoid;
5. the field operations of k can be recovered from the generating morphisms;
6. the generating morphisms (1)-(4) commute with scalar multiplication.

Note that item (2) makes FinRel_k into a $\dagger$-compact category, allowing us to mention the adjoints of generating morphisms in the subsequent relations. Item (5) means that $+$, $\cdot, 0, 1$ and also additive and multiplicative inverses in the field k can be expressed in terms of signal-flow diagrams in the manner we have explained.

3 An example

A famous example in control theory is the 'inverted pendulum': an upside-down pendulum on a cart [11]. The pendulum naturally tends to fall over, but we can stabilize it by setting up a feedback loop where we observe its position and move the cart back and forth in a suitable way based on this observation. Without introducing this feedback loop, let us see how signal-flow diagrams can be used to describe the pendulum and the cart. We shall see that the diagram for a system made of parts is built from the diagrams for the parts, not merely by composing and tensoring, but also with the help of duplication and coduplication, which give additional ways to set variables equal to one another.

Suppose the cart has mass M and can only move back and forth in one direction, so its position is described by a function $x(t)$. If it is acted on by a total force $F_{\text{net}}(t)$ then Newton's second law says

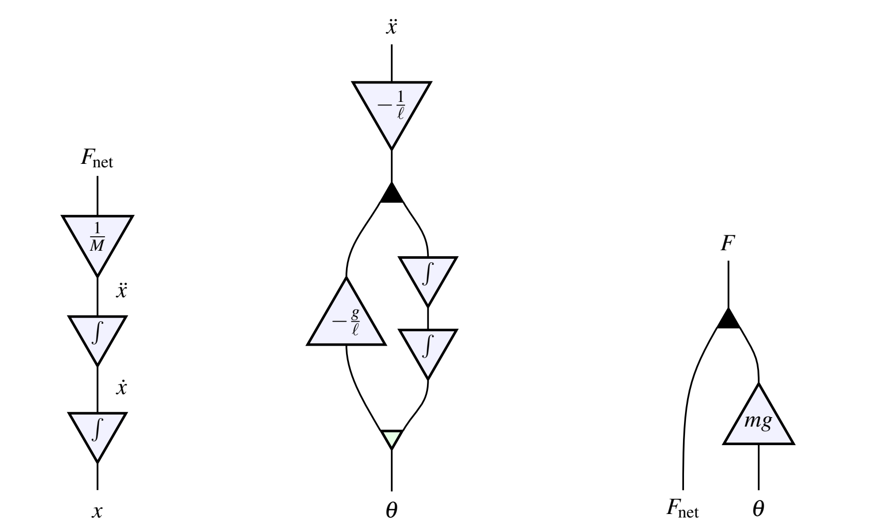
$$F_{\text{net}}(t) = M\ddot{x}(t).$$

We can thus write a signal-flow diagram with the force as input and the cart's position as output, as in Figure 5a.

The inverted pendulum is a rod of length ℓ with a mass m at its end, mounted on the cart and only able to swing back and forth in one direction, parallel to the cart's movement. If its angle from vertical, $\theta(t)$, is small, then its equation of motion is approximately linear:

$$\ell\ddot{\theta}(t) = g\theta(t) - \ddot{x}(t)$$

where g is the gravitational constant. We can turn this equation into a signal-flow diagram with $\ddot{x}$ as input and θ as output, as in Figure 5b. Note that this already includes a kind of feedback loop, since the pendulum's angle affects the force on the pendulum.



(a) The cart alone (b) The pendulum alone (c) The pendulum–cart interface

Figure 5

Finally, there is an equation describing the total force on the cart:

$$F_{\text{net}}(t) = F(t) - mg\theta(t)$$

where $F(t)$ is an externally applied force and $-mg\theta(t)$ is the force due to the pendulum. It will be useful to express this as the signal-flow diagram in Figure 5c. Here we are treating θ as an output rather than an input, implicitly using a cap.

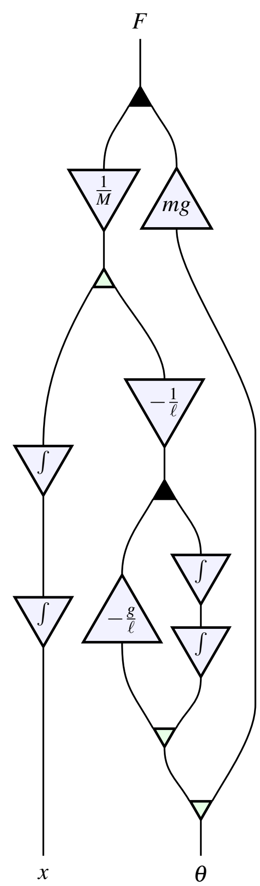
These three signal-flow diagrams above describe the following linear relations:

$$x = \int\int \frac{1}{M} F_{\mathrm{net}} \tag{1}$$

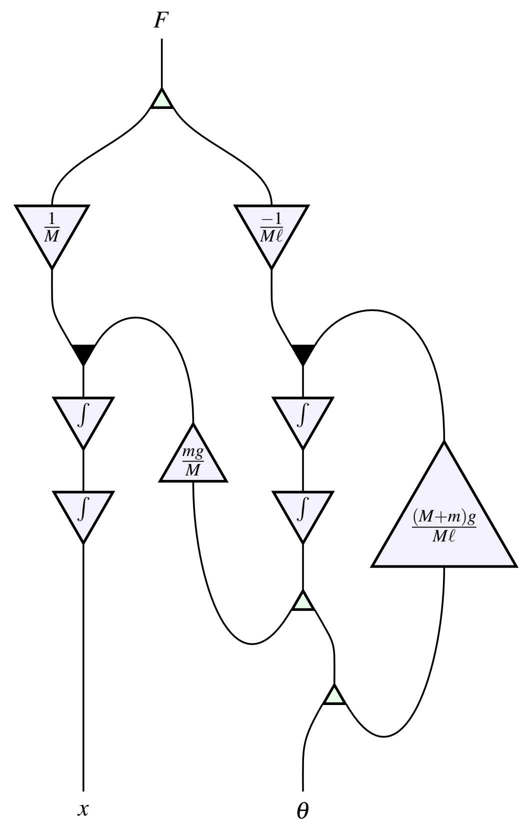
$$\theta = \int\int \left(\frac{g}{\ell}\theta - \frac{1}{\ell}\ddot{x} \right) \tag{2}$$

$$F_{\mathrm{net}} + mg\theta = F \tag{3}$$

where we treat (1) as a relation with F_{net} as input and x as output, (2) as a relation with $\ddot{x}$ as input and θ as output, and (3) as a relation with F as input and $(F_{\mathrm{net}}, \theta)$ as output.



(a) Our inverted pendulum



(b) Friedland's inverted pendulum

Figure 6

To understand how the external force affects the position of the cart and the angle of the pendulum, we wish to combine all three diagrams to form a signal-flow diagram that has the external force F as input and the pair (x, θ) as output. This is not just a simple matter of composing and tensoring the three diagrams. We can take F_{net}, which is an output of (3), and use it as an input for (1). But we also need to duplicate $\ddot{x}$, which appears as an intermediate variable in (1) since $\ddot{x} = \frac{1}{M}F_{\mathrm{net}}$, and use it as an input for (2). Finally, we need to take the variable θ, which appears as an output of both (2) and (3), and identify the two copies of this variable using coduplication. The result is the signal-flow diagram in Figure 6a.

This is not the signal-flow diagram for the inverted pendulum that one sees in Friedland's textbook on control theory [11]. We leave it as an exercise to the reader to rewrite our diagram using the rules given in this paper, obtaining Friedland's diagram in Figure 6b. As a start, one can use Theorem 4 to prove that it is indeed possible to do this rewriting. To do this, simply check that both signal-flow diagrams define the same linear relation. The proof of the theorem in [3] gives a method to actually do the rewriting—but not necessarily the fastest method.

Acknowledgements

We thank Jamie Vicary for pointing out the relevance of the ZX calculus when the first author gave a talk on this material at Oxford in February 2014 [2]. Discussions with Brendan Fong have also been useful. On April 30 of that year, after much of [3] was written, Paweł Sobociński told the first author about some closely related and very interesting papers that he wrote with Filippo Bonchi and Fabio Zanasi [5, 6].

References

[1] Samson Abramsky and Bob Coecke, A categorical semantics of quantum protocols, *Proceedings of the 19th IEEE Conference on Logic in Computer Science (LiCS'04)*, IEEE Computer Science Press, 2004, pp. 415–425, doi:10.1109/LICS.2004.1319636. Available as arXiv:quant-ph/0402130.

[2] John Baez, Network theory I: electrical circuits and signal-flow graphs, lecture at the Department of Computer Science, University of Oxford, February 25, 2014. Slides and video available at http://math.ucr.edu/home/baez/networks_oxford/.

[3] John Baez and Jason Erbele, Categories in control. Available as arXiv:1405.6881.

[4] John Baez and Mike Stay, Physics, topology, logic and computation: a Rosetta Stone, in *New Structures for Physics*, ed. Bob Coecke, *Lecture Notes in Physics* vol. 813, Springer, Berlin, 2011, pp. 95–172, doi:10.1007/978-3-642-12821-9_2. Available as arXiv:0903.0340.

[5] Filippo Bonchi, Paweł Sobociński and Fabio Zanasi, Interacting Hopf algebras. Available as arXiv:1403.7048.

[6] Filippo Bonchi, Paweł Sobociński and Fabio Zanasi, A categorical semantics of signal flow graphs, in *CONCUR 2014–Concurrency Theory*, eds. P. Baldan and D. Gorla, *Lecture Notes in Computer Science* vol. 8704, Springer, Berlin, 2014 pp. 435–450, doi:10.1007/978-3-662-44584-6_30. Also available at http://users.ecs.soton.ac.uk/ps/papers/sfg.pdf.

[7] Aurelio Carboni and Robert F. C. Walters, Cartesian bicategories I, *J. Pure Appl. Alg.* **49** (1987), 11–32, doi:10.1016/0022-4049(87)90121-6.

[8] Bob Coecke and Ross Duncan, Interacting quantum observables: categorical algebra and diagrammatics, *New J. Phys.* **13** (2011), 043016, doi:10.1088/1367-2630/13/4/043016. Available as arXiv:0906.4725.

[9] Bob Coecke and Eric Oliver Paquette, Categories for the practising physicist, in *New Structures for Physics*, ed. Bob Coecke, *Lecture Notes in Physics* vol. 813, Springer, Berlin, 2011, pp. 173–286, doi:10.1007/978-3-642-12821-9_3. Available as arXiv:0905.3010.

[10] Bob Coecke, Dusko Pavlovic and Jamie Vicary, A new description of orthogonal bases, *Math. Str. Comp. Sci.* **23** (2013), 555–567, doi:10.1017/S0960129512000047. Available as arXiv:0810.0812.

[11] Bernard Friedland, *Control System Design: An Introduction to State-Space Methods*, Courier Dover Publications, 2012.

[12] André Joyal and Ross Street, The geometry of tensor calculus I, *Adv. Math.* **88** (1991), 55-113, doi:10.1016/0001-8708(91)90003-P.

[13] André Joyal and Ross Street, The geometry of tensor calculus II. Draft available at http://maths.mq.edu.au/∼street/GTCII.pdf.

[14] Joachim Kock, *Frobenius Algebras and 2D Topological Quantum Field Theories*, Cambridge U. Press, Cambridge, 2003, doi:10.1017/CBO9780511615443. Short version available at http://mat.uab.es/∼kock/TQFT/FS.pdf.

[15] Joachim Kock, Remarks on the origin of the Frobenius equation, available at http://mat.uab.es/∼kock/TQFT.html#history.

[16] Yves Lafont, Towards an algebraic theory of Boolean circuits, *J. Pure Appl. Alg.* **184** (2003), 257–310, doi:10.1016/S0022-4049(03)00069-0. Available at http://iml.univ-mrs.fr/∼lafont/pub/circuits.pdf.

[17] Robert Rosebrugh, Nicoletta Sabadini and Robert F. C. Walters, Generic commutative separable algebras and cospans of graphs, *Th. Appl. Cat.* **15** (2005), 264–277. Available at http://www.tac.mta.ca/tac/volumes/15/6/15-06abs.html.

[18] Peter Selinger, Dagger compact closed categories and completely positive maps, *Elec. Notes Theor. Comp. Sci.* **170** (2007), 139–163, doi:10.1016/j.entcs.2006.12.018.

[19] Jamie Vicary, Categorical formulations of finite-dimensional quantum algebras, *Comm. Math. Phys.* **304** (2011), 765–796, doi:10.1007/s00220-010-1138-0. Available as arXiv:0805.0432.