Universal laws and architectures:
Layering, learning, and decentralized control in neuroscience

John Doyle
John G Braun Professor
Control and Dynamical Systems, EE, BE
Caltech
Outline: Laws and architectures

• Motivating case studies
  – Computers, networks
  – Cells
  – Physiology
  – Brains

• Bits of theory
  – Computation, Turing
  – Control
  – Info theory, stat mech

Emphasis

• Who, what, how, **why**
• Accident versus **necessity**
Turing on layering

The 'skin of an onion' analogy is also helpful. In considering the functions of the mind or the brain we find certain operations which we can explain in purely mechanical terms. This we say does not correspond to the real mind: it is a sort of skin which we must strip off if we are to find the real mind. But then in what remains we find a further skin to be stripped off, and so on. Proceeding in this way do we ever come to the 'real' mind, or do we eventually come to the skin which has nothing in it? In the latter case the whole mind is mechanical.

1950, Computing Machinery and Intelligence, Mind
“Universal laws and architectures?”

- Universal “conservation laws” (constraints)
- Universal architectures (constraints that deconstrain)
- Mention recent papers*
- Focus on broader context not in papers
- Lots of case studies for motivation

*try to get you to read them?
Other case studies (not today)

• Other complex tech nets, aerospace, etc
• Wildfire ecosystems
• Turbulence
• Stat mech foundations

• Synesthesia
Turing (1912-1954)

- Turing 100th birthday in 2012
- Turing
  - machine (math, CS)
  - test (AI, neuroscience)
  - pattern (biology)
- Arguably greatest*
  - all time math/engineering combination
  - WW2 hero
  - “invented” software

*Also world-class runner.
Key papers/results

- Theory (1936): Turing machine (TM), computability, (un)decidability, universal machine (UTM)
- Practical design (early 1940s): code-breaking, including the design of code-breaking machines
- Practical design (late 1940s): general purpose digital computers and software, layered architecture
- Theory (1950): Turing test for machine intelligence
- Theory (1952): Reaction diffusion model of morphogenesis, plus practical use of digital computers to simulate biochemical reactions
Fast and flexible

- Solve problems
- Make decisions
- Take actions
Laws and architectures

Fast

Slow

Flexible

Inflexible

Architecture (constraints that deconstrain)

Architectures (constraints)

laws
Each theory $\approx$ one dimension
Tradeoffs *across* dimensions
Assume architectures a priori
Progress is encouraging, but…
Stovepipes are an obstacle…
Compute
Turing

Delay is most important

Bode

Control, OR

Communicate
Shannon

Delay is least important

Carnot

Boltzmann

Physics

Heisenberg

Einstein
Control, OR

Compute

Turing

Delay is *most* important

Bode

Control, OR
Compute

Turing

Delay is *most* important

Bode

Control, OR
Turing as “new” starting point?

Software
Hardware

Digital
Analog

Compute
Turing

Delay is *most* important

Bode

Control, OR
Turing’s 3 step research:
0. Virtual (TM) machines
1. hard limits, (un)decidability using standard model (TM)
2. Universal architecture achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)

Essentials:
0. Model
1. Universal laws
2. Universal architecture
3. Practical implementation

Turing as “new” starting point?

Software

Hardware

Digital

Analog
Who and what
Neuro motivation

Fast

Inflexible
• Acquire
• Translate/integrate
• Automate

Ashby & Crossley

Slow Flexible

Prefrontal

Motor

Sensory

Striatum

Learning

Thanks to Bassett & Grafton
Ashby & Crossley

- Acquire
- Translate/integrate
- Automate

Slow Flexible

Fast Inflexible

Learning
Fast

Inflexible

Slow

Flexible

Striatum

Learning

Fast

Inflexible
Build on Turing to show what is necessary to make this work.

- Acquire
- Translate/integrate
- Automate

Slow Flexible

Prefrontal Learning Sensory

Motor Striatum

Fast Inflexible

Reflex
Flexible

General purpose
Large uncertainties
Diverse problems

Solve problems
Make decisions
Take actions

Low latency/delay

Fast

Fast
Human complexity

Robust

😊 Metabolism
😊 Regeneration & repair
😊 Healing wound /infect

Fragile

😔 Obesity, diabetes
😔 Cancer
😔 AutoImmune/Inflame

Start with physiology

Lots of triage
Benefits

Robust

😊 Metabolism
😊 Regeneration & repair
😊 Healing wound /infect

😊 Efficient
😊 Mobility
😊 Survive uncertain food supply
😊 Recover from moderate trauma and infection
Mechanism?

Robust

😊 Metabolism
😊 Regeneration & repair
😊 Healing wound /infect

😢 Fat accumulation
😢 Insulin resistance
😢 Proliferation
😊 Inflammation

Fragile

😢 Obesity, diabetes
😢 Cancer
😢 AutoImmune/Inflame

😢 Fat accumulation
😢 Insulin resistance
😢 Proliferation
😢 Inflammation
What’s the difference?

**Robust**
- 😊 Metabolism
- 😊 Regeneration & repair
- 😊 Healing wound / infect

**Fragile**
- 😞 Obesity, diabetes
- 😞 Cancer
- 😞 AutoImmune/Inflame

- 😞 Fat accumulation
- 😞 Insulin resistance
- 😞 Proliferation
- 😞 Inflammation

**Controlled**

**Dynamic**

**Uncontrolled**

**Chronic**
Controlled Dynamic
Low mean
High variability

- Fat accumulation
- Insulin resistance
- Proliferation
- Inflammation
Controlled
Dynamic
Low mean
High variability

Uncontrolled
Chronic
High mean
Low variability

Fat accumulation
Insulin resistance
Proliferation
Inflammation

Death

辛勤

动态
低平均值
高变异性

失控
慢性
高平均值
低变异性

脂肪积聚
胰岛素抵抗
增殖
炎症

死亡

辛勤

动态
低平均值
高变异性

失控
慢性
高平均值
低变异性

脂肪积聚
胰岛素抵抗
增殖
炎症

死亡
Restoring robustness?

**Robust**
- Metabolism
- Regeneration & repair
- Healing wound /infect
  - Fat accumulation
  - Insulin resistance
  - Proliferation
  - Inflammation

**Fragile**
- Obesity, diabetes
- Cancer
- AutoImmune/Inflame
  - Fat accumulation
  - Insulin resistance
  - Proliferation
  - Inflammation

**Controlled**
- Dynamic
- Low mean
- High variability

**Uncontrolled**
- Chronic
- High mean
- Low variability
# Human complexity

<table>
<thead>
<tr>
<th>Robust</th>
</tr>
</thead>
<tbody>
<tr>
<td>😊 Metabolism</td>
</tr>
<tr>
<td>😊 Regeneration &amp; repair</td>
</tr>
<tr>
<td>😊 Immune/inflammation</td>
</tr>
<tr>
<td>😊 Microbe symbionts</td>
</tr>
<tr>
<td>😊 Neuro-endocrine</td>
</tr>
<tr>
<td>☹️ Complex societies</td>
</tr>
<tr>
<td>☹️ Advanced technologies</td>
</tr>
<tr>
<td>☹️ Risk “management”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Yet Fragile</th>
</tr>
</thead>
<tbody>
<tr>
<td>😞 Obesity, diabetes</td>
</tr>
<tr>
<td>😞 Cancer</td>
</tr>
<tr>
<td>😞 AutoImmune/Inflame</td>
</tr>
<tr>
<td>😞 Parasites, infection</td>
</tr>
<tr>
<td>😞 Addiction, psychosis,…</td>
</tr>
<tr>
<td>☠️ Epidemics, war,…</td>
</tr>
<tr>
<td>☠️ Disasters, global &amp;!%$#</td>
</tr>
<tr>
<td>☠️ Obfuscate, amplify,…</td>
</tr>
</tbody>
</table>

**Accident or necessity?**
Robust

😊 Metabolism
😊 Regeneration
😊 Healing wound / infect

Fragile

😊 Obesity, diabetes
😊 Cancer
😊 AutoImmune/Inflame

• Fragility ← Hijacking, side effects, unintended ...
• Of mechanisms evolved for robustness
• Complexity ← control, robust/fragile tradeoffs
• Math: robust/fragile constraints ("conservation laws")

Both

Accident or necessity?
Some features robust to some perturbations

Other features or other perturbations
Some features robust to some perturbations

Other features or other perturbations

Increased complexity?
Robust Modular Simple Plastic Evolvable

and

Fragile Distributed Complex Frozen Frozen

tradeoffs
Horizontal App Transfer
Software
Hardware
Horizontal HW Transfer
Digital
Analog
Cyber-physical: decentralized control with internal delays.
Decentralized, but initially assume computation is fast and memory is abundant.
Plant is also distributed with its own component dynamics
Internal delays between components, and their sensor and actuators, and also externally between plant components.
Going beyond black box: control is decentralized with internal delays.

Huge theory progress in last decade, year, mo., …
The best case study so far

Layered architecture of the bacterial biosphere

Not done here in detail, see slides elsewhere
How?

Universal architectures
Modern technology gives lots of intermediate alternatives.
Want to emphasize the differences between these two types of layering.
What matters is the OS.
• Some people write apps and build hardware
• But most software and hardware is acquired by “horizontal” transfer from others

• Similarly, most new ideas (humans) and new genes (bacteria) are acquired horizontally
“solution sets” (a la Marder, Prinze, etc)

large, thin, nonconvex

All systems

Software

Hardware

Digital

Analog

Functional
Letters and words

- 9 letters: adeginorz
- $9! = 362,880$ sequences of 9 letters
- Only “organized” is a word

$1 \ll (\# \text{ words}) \ll (\# \text{ non-words})$

large thin
Computer programs

• Almost any computer language
• Large # of working programs
• Much larger # of non-working programs
• “Nonconvex” = simple mashups of working programs don’t work

1  \ll  (# programs)  \ll  (# non-programs)

large  thin
large thin

1 << # toys << # piles

toys

pile
edge of chaos
self-organized criticality
scale-free

statistical physics
random ensembles
minimally tuned
phase transitions
bifurcations

“order for free?”
large thin
1 \ll \# \text{toys} \ll \# \text{piles}

nonconvex

“order for free?”
large, thin, nonconvex

All “code”

functional software

digital

Analog

Software

Hardware

Digital

Analog
This paper aims to bridge progress in neuroscience involving sophisticated quantitative analysis of behavior, including the use of robust control, with other relevant conceptual and theoretical frameworks from systems engineering, systems biology, and mathematics.

Architecture, constraints, and behavior

John C. Doyle\textsuperscript{a,1} and Marie Csete\textsuperscript{b,1}

\textsuperscript{a}Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125; and \textsuperscript{b}Department of Anesthesiology, University of California, San Diego, CA 92103

Edited by Donald W. Pfaff, The Rockefeller University, New York, NY, and approved June 10, 2011 (received for review March 3, 2011)

This paper aims to bridge progress in neuroscience involving sophisticated quantitative analysis of behavior, including the use of robust control, with other relevant conceptual and theoretical frameworks from systems engineering, systems biology, and mathematics. Familiar and accessible case studies are used to illustrate concepts of robustness, organization, and architecture (modularity and protocols) that are central to understanding complex networks. These essential organizational features are hidden during normal function of a system but are fundamental for understanding the nature, design, and function of complex biologic and technologic systems.

Doyle, Csete, Proc Nat Acad Sci USA, JULY 25 2011
- Acquire
- Translate/integrate
- Automate
Flexible/Adaptable/Evolvable

Horizontal Meme Transfer

Sensory

Prefrontal

Striatum

Learning

Reflex

Catabolism

AA

RN

transl.

Proteins

xRNA transc

Precursors

DNA

Repl.

Gene

ATP

ATP

Ribosome

RNAp

D

NAp

Software

Hardware

Horizontal App Transfer

Horizontal Gene Transfer

Digital

Analog

Depends crucially on layered architecture
Most
• software and hardware
• new ideas (humans)
• new genes (bacteria)

is acquired by “horizontal” transfer, though sometimes it is evolved locally.
Sequence ~100 E Coli (not chosen randomly)
- ~ 4K genes per cell
- ~20K different genes in total
- ~1K universally shared genes
Exploiting layered architecture

Horizonal Bad Gene Transfer

Virus

Horizontal Bad App Transfer

Virus

Horizontal Bad Meme Transfer

Fragility?

Parasites & Hijacking
Build on Turing to show what is necessary to make this work.

- Acquire
- Translate/integrate
- Automate

Horizontal Meme Transfer

Horizontal App Transfer

Horizontal Gene Transfer

Amazingly Flexible/Adaptable

Depends crucially on layered architecture
Delay is even more important

Universal laws and architectures

Control

Bode

Turing

Compute

Plant

Act

Sense

Control

Slow Flexible

Hardware

Software

Digital

Analog

Fast Inflexible
Why

Necessity

Why

Turing

Compute
Turing’s 3 step research:
0. **Virtual (TM) machines**
1. hard limits, (un)decidability using standard model (TM)
2. Universal architecture achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)

Essentials:
0. **Model**
1. Universal laws
2. Universal architecture
3. Practical implementation
• ...being digital should be of greater interest than that of being electronic. That it is electronic is certainly important because these machines owe their high speed to this... But this is virtually all that there is to be said on that subject.
• That the machine is digital however has more subtle significance. ... One can therefore work to any desired degree of accuracy.

1947 Lecture to LMS
• ... digital ... of greater interest than that of being electronic ...
• ...any desired degree of accuracy...
• This accuracy is not obtained by more careful machining of parts, control of temperature variations, and such means, but by a slight increase in the amount of equipment in the machine.

1947 Lecture to LMS
Summarizing Turing:
- Digital more important than electronic...
- Robustness: accuracy and repeatability.
- Achieved more by internal hidden complexity than precise components or environments.
• … quite small errors in the initial conditions can have an overwhelming effect at a later time. The displacement of a single electron by a billionth of a centimetre at one moment might make the difference between a man being killed by an avalanche a year later, or escaping.

1950, Computing Machinery and Intelligence, *Mind*
... quite small errors in the initial conditions can have an overwhelming effect at a later time....

- It is an essential property of the mechanical systems which we have called 'discrete state machines' that this phenomenon does not occur.
- Even when we consider the actual physical machines instead of the idealised machines, reasonably accurate knowledge of the state at one moment yields reasonably accurate knowledge any number of steps later.

1950, Computing Machinery and Intelligence, *Mind*
Turing’s 3 step research:
0. Virtual (TM) machines
1. **hard limits, (un)decidability using standard model (TM)**
2. Universal architecture achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)
Logic

slow

time

fast

large space

TM has $\infty$ memory

$\infty$ memory
Logic

\[ \infty \text{ memory} \]

- Time
  - Fast
  - Slow

- Space
  - Large space
  - Space is free

- TM has \( \infty \text{ memory} \)
Decidable problem $= \exists$ algorithm that solves it

Most naively posed problems are undecidable.
Turing’s 3 step research:
0. Virtual (TM) machines
1. hard limits, (un)decidability using standard model (TM)
2. **Universal architecture** achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)
2. **Universal architecture achieving hard limits (UTM)**

- Software: A Turing machine (TM) can be data for another Turing machine.
- A Universal Turing Machine can run any TM.
- A UTM is a virtual machine.
- There are lots of UTMs, differ only (but greatly) in speed and programmability (space assumed free).
The halting problem

- Given a TM (i.e. a computer program)
- Does it halt (or run forever)?
- Or do more or less anything in particular.
- Undecidable! There does not exist a special TM that can tell if any other TM halts.
- i.e. the program HALT does not exist. 😞
\textbf{Thm:} TM H=HALT does not exist.

That is, there does not exist a program like this:

\[ H(TM, input) \triangleq \begin{cases} 
1 \text{ if } TM(input) \text{ halts} \\
0 \text{ otherwise} 
\end{cases} \]

\textbf{Proof} is by contradiction. Sorry, don’t know any alternative. And Turing is a god.
Thm: No such \( H \) exists.

Proof: Suppose it does. Then define 2 more programs:

\[
H(TM, input) \triangleq \begin{cases} 
1 & \text{if } TM(input) \text{ halts} \\
0 & \text{otherwise}
\end{cases}
\]

\[
H'(TM, input) \triangleq \begin{cases} 
1 & \text{if } H(TM, input) = 0 \\
\text{loop forever} & \text{otherwise}
\end{cases}
\]

\[
H*(TM) \triangleq H'(TM, TM)
\]

Run \( H*(H*) = H'(H*, H*) \)

\[
= \begin{cases} 
\text{halt if } H*(H*) \text{ loops forever} \\
\text{loop forever otherwise}
\end{cases}
\]

Contradiction!
Implications

• Large, thin, nonconvex everywhere...
• TMs and UTMs are perfectly repeatable
• But perfectly unpredictable
• Undecidable: Will a TM halt? Is a TM a UTM? Does a TM do X (for almost any X)?
• Easy to make UTMs, but hard to recognize them.
• Is anything decidable? Yes, questions NOT about TMs.
Fast
Slow
Really slow

Inflexible/
Specific
Undecidable
Decidable
NP
P

Flexible/
General

Hard limits

Computational complexity
Flexible/
General

Inflexible/
Specific

Undecidable

Intrinsic
complexity
classes

Decidable

Computational
complexity

NP

Flexible/General

Inflexible/Specific

P
These are hard limits on the *intrinsic* computational complexity of *problems*.

Must still seek algorithms that achieve the limits, and architectures that support this process.
Delay is even more important in control.

Computational complexity of:
- Designing control algorithms
- Implementing control algorithms
Slow
Flexible

Fast
Flexible

Unachievable robustness

Fast
Inflexible

Most UTMs here

Hopeless fragility

Hard limits
Issues for engineering

• Turing remarkably relevant for 76 years
• UTMs are $\approx$ implementable
  – Differ only (but greatly) in speed and programmability
  – Time/speed/delay is most critical resource
  – Space (memory) almost free for most purposes
• Read/write random access memory hierarchies
• Further gradations of decidable (P/NP/coNP)
• Most crucial:
  – UTMs differ vastly in speed, usability, and programmability
  – You can fix bugs but it is hard to automate finding/avoiding them
Issues for engineering

- Turing remarkably relevant for 76 years
- UTMs are \( \approx \) implementable
  - Differ only (but greatly) in speed and programmability
  - Time/speed/delay is most critical resource
  - Space (memory) almost free for most purposes
- Read/write random access memory hierarchies
- Further gradations of decidable (P/NP/coNP)
- **Most crucial:**
  - UTMs differ *vastly* in speed, usability, and programmability
  - You can fix bugs but it is hard to automate finding/avoiding them
Conjectures, biology

- Memory potential \( \approx \infty \)
- Examples
  - Insects
  - Scrub jays
  - Autistic Savants

- But why so rare and/or accidental?
- Large memory, computation of limited value?
- Selection favors fast robust action?
- Brains are distributed (not studied by Gallistel)
• Suppose we only care about space?
• And time is free
• Bad news: compression undecidable.
• Shannon: change the problem!
Shannon’s brilliant insight

- Forget time
- Forget files, use *infinite random ensembles*

**Good news**

- Laws and architecture!
- Info theory most popular and accessible topic in systems engineering
- *Fantastic* for some engineering problems
Shannon’s brilliant insight
• Forget time
• Forget files, use *infinite random ensembles*

**Bad news**
• Laws and architecture very brittle

• Less than zero impact on internet architecture
• Almost useless for biology (But see Lestas et al, 2010)
• Misled, distracted generations of biologists (and neuroscientists)
Delay is *most* important

New progress!

Lowering the barrier

Delay is *least* important

Compute
- Turing
- Bode
- Control, OR

Communicate
- Shannon
- Carnot
- Boltzmann
- Physics

Shannon
- Turing
- Bode
- Control, OR

Shannon
- Carnot
- Boltzmann
- Physics

Einstein
- Control, OR
- Heisenberg
- Compute
- Communicate

Einstein
- Compute
- Communicate

Einstein
- Compute
- Communicate

Einstein
- Compute
- Communicate

Einstein
- Compute
- Communicate
Delay is *even more* important

- Acquire
- Translate/integrate
- Automate

Wolpert, Grafton, etc

**robust**

Brain as optimal controller

Reflex
\[ x_{t+1} = p x_t + w_t + u_{t-a} \]

\[ p > 1 \]
delay $a$

\[ u_{t-a} = -(px_t + w_t) \]

\[ \Rightarrow \|x\| \approx 0 \quad \|u\| \approx \|w\| \]

$\|x\| \approx 0$  \quad \|u\| \approx \|w\|$

\[ x_{t+1} = px_t + w_t + u_{t-a} \]

$p > 1$
No delay or no uncertainty

\[ u_{t-a} = - (px_t + w_t) \]

\[ \Rightarrow \|x\| \approx 0 \quad \|u\| \approx \|w\| \]

With delay \textit{and} uncertainty

\[ x_{t+1} = px_t + w_t + u_{t-a} \]

\[ p > 1 \]

\[ \Rightarrow \|x\| \approx \|u\| \approx p^a \|w\| \]
Linearized pendulum on a cart

\[
\begin{align*}
\frac{d}{dt} & \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \\
& \begin{bmatrix}
0 & 0 & \frac{1}{q} & 0 \\
0 & \frac{m^2gl^2}{q} & 0 & \frac{-(J + ml^2)b}{q} \\
0 & \frac{mgl(M + m)}{q} & 0 & \frac{-mlb}{q} \\
0 & 0 & 0 & 0
\end{bmatrix} x + \\
& \begin{bmatrix}
0 \\
0 \\
0 \\
\frac{J + ml^2}{q} \\
\frac{ml}{q}
\end{bmatrix} u
\end{align*}
\]

\[q = J(M + m) + Mml^2\]
\((M + m)\ddot{x} + ml(\ddot{\theta}\cos\theta - \dot{\theta}^2 \sin\theta) = u\)

\(\ddot{x}\cos\theta + l\ddot{\theta} + g\sin\theta = 0\)

\(y = x + \alpha l \sin\theta\)

linearize

\((M + m)\ddot{x} + ml\dot{\theta} = u\)

\(\ddot{x} + l\ddot{\theta} \pm g\theta = 0\)

\(y = x + \alpha l \theta\)
Robust
= agile and balancing
Robust
= agile and balancing
Efficient = length of pendulum (artificial)
\[
\begin{bmatrix}
x \\
\theta
\end{bmatrix} = \frac{1}{D(s)} \begin{bmatrix}
ls^2 \pm g \\
-s^2
\end{bmatrix} u
\]

\[D(s) = s^2 \left( Mls^2 \pm (M + m)g \right)\]

\[y = x + \alpha l \theta = \frac{\varepsilon ls^2 \pm g}{D(s)}\]

\[\varepsilon = 1 - \alpha\]

\[p = \sqrt{\frac{g}{l}} \sqrt{1+r} \quad r = \frac{m}{M}\]

\[z = \sqrt{\frac{g}{l}} \sqrt{\frac{1}{\varepsilon}}\]

\[\begin{aligned}
(M + m) \ddot{x} + ml\ddot{\theta} &= u \\
\ddot{x} + l\ddot{\theta} \pm g\theta &= 0 \\
y &= x + \alpha l \theta
\end{aligned}\]
Delay $\tau$

$p \propto \sqrt{\frac{1}{l}}$

$|T(j\omega)| = \left| \frac{E}{N} \right|$

error

noise
\[ \frac{1}{\pi} \int_0^\infty \ln |T(j\omega)| \, d\omega \geq 0 \]

Easy, even with eyes closed
No matter what the length

Proof: Standard UG control theory:
Easy calculus, easier contour integral, easiest Poisson Integral formula
Harder if delayed or short

Delay

Short

$\varepsilon l$

$M$
Also harder if sensed low

\[ r = \frac{m}{M} \]
Delay $\tau$

$$p \propto \sqrt{\frac{1}{l}}$$

$$|T(j\omega)| = \left| \frac{E}{N} \right|$$
Delay is hard

\[
\frac{1}{\pi} \int_0^\infty \ln |T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq \ln |T_{mp}(p)| = p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
Delay $\tau$ is hard for any controller so is an intrinsic constraint on the difficulty of the problem.

\[
\frac{1}{\pi} \int_{0}^{\infty} \ln|T(j\omega)| \frac{2p}{p^{2} + \omega^{2}} d\omega \geq \ln|T_{mp}(p)| = p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
\[
\frac{1}{\pi} \int_0^\infty \ln|T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq p\tau \propto \tau \sqrt{\frac{1}{l}}
\]

For fixed length

Fragility

Too fragile

\[\tau \sqrt{\frac{1}{l}}\]

up

down

large \(\tau\)
small \(1/\tau\)
small \(\tau\)
large \(1/\tau\)

1/delay

L
\[
\frac{1}{\pi} \int_{0}^{\infty} \ln|T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq p\tau \propto \tau \sqrt{\frac{1}{l}}
\]

We would like to tolerate large delays (and small lengths), but large delays severely constrain the achievable robustness.

large $\tau$ \hspace{2cm} small $\tau$

small $1/\tau$ \hspace{2cm} large $1/\tau$
\[
\frac{1}{\pi} \int_{0}^{\infty} \ln |T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq \ln |T_{mp}(p)| = p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
Fragility

\[
\frac{1}{\pi} \int_{0}^{\infty} \ln|T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq \ln|T_{mp}(p)| = p\tau \propto \tau \sqrt{\frac{1}{l}}
\]

For fixed delay

Why oscillations?
Side effects of hard tradeoffs

Fragility

\[ p \propto \sqrt{\frac{1}{l}} \]

Too fragile
The ratio of delay between people is proportional to the lengths they can stabilize.

\[
\frac{1}{\pi} \int_0^\infty \ln |T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq \ln |T_{mp}(p)| = p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
Eyes moved down is harder (RHP zero)
Similar to delay
Suppose \( r = \frac{m}{M} \ll 1 \)

Units \( \Rightarrow M = g = 1 \)

\[
y = x + \alpha l \theta = \frac{\varepsilon ls^2 \pm g}{s^2 (ls^2 \pm g)} \quad \varepsilon = 1 - \alpha
\]

\[
p \approx \sqrt{\frac{g}{l}} \quad z = \sqrt{\frac{g}{l} \frac{1}{\varepsilon}} \Rightarrow \frac{z + p}{z - p} = \frac{1 + \sqrt{\varepsilon}}{1 - \sqrt{\varepsilon}}
\]
Compare

\[ p = \sqrt{\frac{g}{l(1-\varepsilon)}} \sqrt{1+r} = p_0 \sqrt{\frac{1}{(1-\varepsilon)}} \approx p_0 \left(1 + \frac{\varepsilon}{2}\right) \]

Move eyes

\[ p = \sqrt{\frac{g}{l}} \sqrt{1+r} \quad r = \frac{m}{M} \quad z = \sqrt{\frac{g}{l}} \sqrt{\frac{1}{\varepsilon}} \]

\[ p = z \Rightarrow 1+r = \frac{1}{\varepsilon} \Rightarrow \varepsilon = \frac{1}{1+r} \]

\[ p \left(1+\frac{1}{3}\frac{p^2}{z^2}\right) = \sqrt{\frac{g}{l}} \sqrt{1+r} \left(1+\frac{1}{3}\varepsilon\right) = p \left(1+\frac{\varepsilon}{3}\right) \]

\[ = p \left(1+\frac{1-\alpha}{3}\right) \]
\[
\frac{1}{\pi} \int_0^\infty \ln |S(j\omega)| \left( \frac{2z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right| \\
\frac{1}{\pi} \int_0^\infty \ln |T(j\omega)| \left( \frac{2p}{p^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right|
\]

\[\varepsilon = \frac{1}{1 + r}\]

\[\frac{z + p}{z - p} = \frac{1 + \sqrt{\varepsilon}}{1 - \sqrt{\varepsilon}}\]

This is a cartoon, but can be made precise.
Hard limits on the **intrinsic** robustness of control **problems**.

Must (and do) have algorithms that achieve the limits, and architectures that support this process.

\[
\frac{1}{\pi} \int_0^\infty \ln |S(j\omega)| \left( \frac{2z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right|
\]

This is a cartoon, but can be made precise.
How do these two constraints (laws) relate?

\[
\frac{1}{\pi} \int_{0}^{\infty} \ln |S(j\omega)| \left( \frac{2z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right|
\]
Delay comes from sensing, communications, computing, and actuation. Delay limits robust performance.

\[
\frac{1}{\pi} \int_0^\infty \ln |T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq \ln |T_{mp}(p)| = p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
This is about speed and flexibility of computation.

How do these two constraints (laws) relate?

Computation delay adds to total delay.

Computation is a component in control.

Delay $\tau$

Noise

Control

small delay

large delay

Flexible

Inflexible

This is about speed and flexibility of computation.
Delay makes control hard.

Computation delay adds to total delay.

Computation is a component in control.

\[
\frac{1}{\pi} \int_0^\infty \ln |T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
How general is this picture?


- fragile
- robust

- simple tech
- complex tech

- efficient
- wasteful
I recently found this paper, a rare example of exploring an explicit tradeoff between robustness and efficiency. This seems like an important paper but it is rarely cited.
Phage lifecycle

- Lyse
- Multiply
- Survive
- Infect
Survive

- fragile
- robust

Multiply

- fast
- slow

Good architectures?

Hard limits?

Capsid Genome

- thin small
- thick big
Chandra, Buzi, and Doyle

Most important paper so far.
Theorem!

\[ \frac{1}{\pi} \int_0^\infty \ln |S(j\omega)| \left( \frac{z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right| \]

\( z \) and \( p \) functions of enzyme complexity and amount

---

Fragility

\[ \ln \left| \frac{z + p}{z - p} \right| \]

---

Simple enzyme

Complex enzyme

---

Enzyme amount
Inside every cell

Almost

Catabolism

Precursors

ATP

AA

Nucl.

Building Blocks

Enzymes

Proteins

Ribosome

RNA

transl.

xRNA

DNA

transc.

Repl.

Gene

RNAp

DNAp

Crosslayer autocatalysis

Macro-layers
Yeast anaerobic glycolysis

Catabolism

Precursors → ATP → Energy

Rest of cell

Minimal model
ATP → Autocatalytic feedback

Energy

Catabolism → ATP

Reaction 1 ("PFK") → Reaction 2 ("PK") → Rest of cell

Yeast anaerobic glycolysis

Minimal model
Robust = Maintain energy (ATP concentration) despite demand fluctuation.

Tight control creates “weak linkage” between power supply and demand.
Constrained ("conserved"): Moieties

1. NAD
2. Adenylate
3. Carbon
4. phosphate
5. oxygen

6. Oxidized state of metabolites
7. Reduced state of metabolites
8. High energy potential release

\[
\frac{1}{\pi} \int_{0}^{\infty} \ln \left| S(j\omega) \right| \left( \frac{z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right| + \ln \left| \frac{H_2O}{H_2O} \right| + \ln \left| \frac{ATP}{ATP} \right| + \ln \left| \frac{ADP}{ADP} \right| + 2 \ln \left| \frac{AMP}{AMP} \right| + \ln \left| \frac{PYR}{PYR} \right| + [ADP] + 2[AMP] + [H_2O]
\]
Hard tradeoff in glycolysis

Fragile

Robust

Robust = Maintain energy (ATP concentration) despite demand fluctuation
What makes this hard?
1. Instability (autocatalysis)
2. Delay (enzyme amount)

Robust
≈ Disturbance rejection
≈ Accurate
What makes this hard?

1. Instability
2. Delay

The CNS must cope with both

Today’s important point
ATP

Reaction
1 ("PFK")

Reaction
2 ("PK")

enzymes catalyze reactions

energy

ATP

Rest of cell
Efficient = low metabolic overhead
≈ low enzyme amount
Reaction rates $\propto$ enzyme amount

Can't make too many enzymes here, need to supply rest of the cell.

Efficient = low metabolic overhead
$\approx$ low enzyme amount
($\Rightarrow$ slow reactions)

Enzymes catalyze reactions, another source of autocatalysis.
Robust = Maintain ATP

Efficient = low enzyme amount
(⇒ slow reactions)
Hard tradeoff in glycolysis is

- robustness vs efficiency
- absent without autocatalysis
- too fragile with simple control
- plausibly robust with complex control

\[
\frac{1}{\pi} \int_0^\infty \ln |S(j\omega)| \left( \frac{z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right|
\]
What (some) reviewers say

• “...to establish universality for all biological and physiological systems is simply wrong. It cannot be done..."

• ... a mathematical scheme without any real connections to biological or medical...

• ...universality is well justified in physics... for biological and physiological systems ...a dream that will never be realized, due to the vast diversity in such systems.

• ...does not seem to understand or appreciate the vast diversity of biological and physiological systems...

• ...a high degree of abstraction, which ...make[s] the model useless ...
This picture is very general


- fragile
  - simple tech
- robust
  - complex tech

cheap
- large delay $\tau$
- fast multiply

metabolic expensive
- small $\tau$
- slow
This picture is very general

<table>
<thead>
<tr>
<th>Domain specific costs/tradeoffs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>metabolic overhead</strong></td>
</tr>
<tr>
<td><strong>CNS reaction</strong> time $\tau$ (delay)</td>
</tr>
<tr>
<td><strong>phage multiplication rate</strong></td>
</tr>
</tbody>
</table>
This picture is very general

Domain specific costs/tradeoffs

- Metabolic cost
- Reaction time $\tau$
- Phage x rate

- Fragile
- Robust

- Simple tech
- Complex tech

- Cheap ↔ expensive
- Large $\tau$ ↔ small $\tau$
- Fast ↔ slow
fragile

Survive

robust

thin small

Capsid thickness Genome size

good architectures?

hard limits?

fast multiply  slow
\[ \frac{1}{\pi} \int_{0}^{\infty} \ln|T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq p\tau \propto \tau \sqrt{\frac{1}{l}} \]
\[
\frac{1}{\pi} \int_0^\infty \ln |S(j\omega)| \left( \frac{2z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right|
\]

\[
\varepsilon = \frac{1}{1 + r}
\]

This is a cartoon, but can be made precise.
Hard tradeoff in glycolysis is
• robustness vs efficiency
• absent without autocatalysis
• too fragile with simple control
• plausibly robust with complex control

Simple, but too fragile

\[
\frac{1}{\pi} \int_{0}^{\infty} \ln |S(j\omega)| \left( \frac{z}{z^2 + \omega^2} \right) d\omega \geq \ln \left| \frac{z + p}{z - p} \right|
\]
Computational complexity

Really slow
Slow
Fast

Fast
Slow
Really slow

Decidable
NP
P

Flexible/General
Inflexible/Specific

Turing has the original “universal law”

Hard limits
Delay makes control hard.

Computation delay adds to total delay.

Computation is a component in control.

\[
\frac{1}{\tau} \int_0^\infty \ln |T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
Fragility

This needs formalization:

What **flexibility** makes control hard?

Large, structured uncertainty?

\[
\frac{1}{\pi} \int_0^\infty \ln |T(j\omega)| \frac{2p}{p^2 + \omega^2} d\omega \geq p\tau \propto \tau \sqrt{\frac{1}{l}}
\]
What about: Cyber-physical: decentralized control with internal delays?
No delay or no uncertainty

\[ u_{t-a} = -\left( p x_t + w_t \right) \]

\[ \Rightarrow \| x \| \approx 0 \quad \| u \| \approx \| w \| \]

With delay and uncertainty

\[ x_{t+1} = p x_t + w_t + u_{t-a} \]

\[ \Rightarrow \| x \| \approx \| u \| \approx p^a \| w \| \]

\[ p > 1 \]
Focus on delays

$$x_{t+1} = px_t + w_t + u_{t-a}$$
$$p > 1$$
Focus on delays

Actuator delay
Decentralized control

t = transmission

l = internal

r = internal

s = sense

a = act

p = plant
\[ l + (p + a + s) + r \]

\( l = \text{internal} \)

\( s = \text{sense} \)

\( r = \text{internal} \)

\( a = \text{act} \)

\( p = \text{plant} \)

\text{total remote + plant delay}
Communications delay

$t=\text{transmission}$
\[ t \leq l + (p + a + s) + r \]

Then decentralized control design can be made **convex**
A primary driver of human brain evolution?

\[ t \leq (p + a + s) \]

\((a + s)\) small
Wolpert, Grafton, etc

robust

Brain as optimal controller

- Acquire
- Translate/integrate
- Automate

Reflex
Going beyond black box: control is decentralized with internal delays.

Huge theory progress in last decade, year, mo., ...
Decentralized, but initially assume computation is fast and memory is abundant.
Plant is also distributed with its own component dynamics.
Internal delays between brain components, and their sensor and actuators, and also externally between plant components
Internal delays involve both computation and communication latencies
This progress is important.

New progress!

Delay is most important.

Delay is least important.

Compute
- Turing
- Control, OR

Communicate
- Shannon
- Bode
- Physics

Control
- Compute
- Communicate
Going beyond black box: control is decentralized with internal delays.

Huge theory progress in last decade, year, mo., …
Going beyond black box: control is decentralized with internal delays.

Mammal NS seems organized to reduce delays in motor control.
Universal architectures

Implications

(Layered architectures discussed elsewhere)
Turing as “new” starting point?

Essentials:
0. Model
1. Universal laws
2. Universal architecture
3. Practical implementation

Turing’s 3 step research:
0. Virtual (TM) machines
1. hard limits, (un)decidability using standard model (TM)
2. Universal architecture achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)
What matters is the OS.
Horizontal Gene Transfer

Horizontal App Transfer

Software

Hardware

Horizontal Meme Transfer

Depends crucially on layered architecture
Sequence ~100 E Coli (*not* chosen randomly)
- ~ 4K genes per cell
- ~20K *different* genes in total
- ~ 1K universally shared genes
selection + drift + mutation + gene flow
+ facilitated variation

Horizontal Gene Transfer
plasmid
virus
HGT
large functional changes in genomes
Genes

natural selection + genetic drift + mutation + gene flow
+ facilitated variation

Genome can have large changes
natural selection + genetic drift + mutation + gene flow + facilitated variation

Small gene change can have large but functional phenotype change
natural selection + genetic drift + mutation + gene flow + facilitated variation

Only possible because of shared, layered, network architecture
Standard theory:
natural selection + genetic drift
+ mutation + gene flow

Greatly abridged cartoon here

Shapiro explains well what this is and why it’s incomplete (but Koonin is more mainstream)
Standard theory:
selection + drift + mutation + gene flow
Standard theory:
selection + drift + mutation + gene flow

No new laws.
No architecture.
No biology.
selection +
drift +
mutation +
gene flow

Phenotype
Gene alleles

Selection

All complexity is emergent from random ensembles with minimal tuning.
No new laws.
No architecture.
The battleground

Phenotype

No gap. Just physics.

Huge gap. Need supernatural

Gene alleles

Genes?
What they agree on

No new laws.
No architecture.
No biology.

Pheno-type

No gap.

Gene alleles

Huge gap.

Genes
Depends crucially on layered architecture

Amazingly Flexible/Adaptable

Horizontal Gene Transfer

Horizontal App Transfer

Horizontal Meme Transfer

Hardware

Software

Digital

Analog
Putting biology back into evolution
Universal architectures

What can go wrong?
Want to emphasize the differences between these two types of layering.
Diverse applications (HMT)
caltech.edu?

IP addresses interfaces (not nodes)

Global and direct access to physical address!

131.215.9.49
Global and direct access to physical address!

Robust?
- Secure
- Scalable
- Verifiable
- Evolvable
- Maintainable
- Designable
- …

IP addresses interfaces (not nodes)
Naming and addressing need to be
• resolved within layer
• translated between layers
• not exposed outside of layer

Related “issues”
• VPNs
• NATS
• Firewalls
• Multihoming
• Mobility
• Routing table size
• Overlays
• …
Until late 1980s, no congestion control, which led to “congestion collapse”
Original design challenge?

TCP/IP

Deconstrained (Applications)

Deconstrained (Hardware)

Constrained

Facilitated wild evolution
Created
• whole new ecosystem
• completely opposite

Networked OS

• Expensive mainframes
• Trusted end systems
• Homogeneous
• Sender centric
• Unreliable comms
Reactions

Assembly

DNA/RNA

cross-layer control

• Highly organized
• Naming and addressing

Cross-layer control

Coming later: contrast with cells

Protein

Assembly
control

Outside

Inside

DNA

Ligands & Receptors

Transmitter

Receiver

Responses

control

control
Next layered architectures

Deconstrained (Applications)

Deconstrained (Hardware)

Constrained

? 

Control, share, virtualize, and manage resources

Few global variables

Don’t cross layers

Comms
Memory, storage
Latency
Processing
Cyber-physical
Persistent errors and confusion (“network science”)

Every layer has different diverse graphs.

Architecture is least graph topology.

Architecture facilitates arbitrary graphs.
The "robust yet fragile" nature of the Internet

John C. Doyle*, David L. Alderson*, Lun Li*, Steven Low*, Matthew Roughan†, Stanislav Shalunov§, Reiko Tanaka¶, and Walter Willinger||

*Engineering and Applied Sciences Division, California Institute of Technology, Pasadena, CA 91125; †Applied Mathematics, University of Adelaide, South Australia 5005, Australia; §Internet2, 3025 Boardwalk Drive, Suite 200, Ann Arbor, MI 48108; ¶Bio-Mimetic Control Research Center, Institute of Physical and Chemical Research, Nagoya 463-0003, Japan; and ||AT&T Labs–Research, Florham Park, NJ 07932

Edited by Robert M. May, University of Oxford, Oxford, United Kingdom, and approved August 29, 2005 (received for review February 18, 2005)

The search for unifying properties of complex networks is popular, challenging, and important. For modeling approaches that focus on no self-loops or parallel edges) having the same graph degree.

We will say that graphs \( g \in G(D) \) have scaling-degree sequen...
Mathematics and the Internet: A Source of Enormous Confusion and Great Potential

Walter Willinger, David Alderson, and John C. Doyle
Unfortunately, not intelligent design

YOUR INNER FISH
A JOURNEY INTO THE 3.5-BILLION-YEAR HISTORY OF THE HUMAN BODY

NEIL SHUBIN

Ouch.
Why?

left recurrent laryngeal nerve
Why? Building humans from fish parts.

FIGURE 3-11 Schematic diagram showing the relationship between the vagus cranial nerve and the arterial arches in fish (a) and human (b). Only the third, fourth, and part of the sixth arterial arches remain in placental mammals, the sixth acting only during fetal development to carry blood to the placenta. The fourth vagal nerve in mammals (the recurrent laryngeal nerve) loops around the sixth arterial arch just as it did in the original fishlike ancestor, but must now travel a greater distance since the remnant of the sixth arch is in the thorax.
It could be worse.
delay = death

sense

move

Spine

Control Loop
Feed-Back Differential

Ascending Neural Radiations to the Hippocampus/Thalamus/hypothalamus
Cerebral Cortex
Anterior Thalamic Nucleus
Cerebral Hemisphere
Olfactory Bulb
Visual Impulses
Hypothalamus
Pituitary Gland
Mamillary Body of Hypothalamus
Amygdaloid Nucleus
Auditory Impulses
Projection to Spinal Cord
Ascending Sensory Tracts

Corpus Callosum
Thalamus
Pineal Gland
Hippocampus
Cerebellum
Vestibulo-ocular reflex

1. Detection of rotation

2. Inhibition of extraocular muscles on one side.

2. Excitation of extraocular muscles on the other side

3. Compensating eye movement
Same actuators
Delay is limiting

Move head
Sense
Fast
Act

Move hand
Sense
Slow
Act

Fast?

Slow
Versus standing on one leg
- Eyes open vs closed
- Contrast
  - young surfers
  - old football players
delay = death

sense
move

Control Loop
Feed-Back Differential

Ascending Neural Radiations to the Hippocampus/Thalamus/hypothalamus

Corpus Callosum
Thalamus
Pineal Gland
Hippocampus

Cerebellum
Auditory Impulses
Projection to Spinal Cord

Ascending Sensory Tracts

Hypothalamus
Pituitary Gland
Mammillary Body of Hypothalamus
Amygdaloid Nucleus

Cerebral Hemisphere
Olfactory Bulb
Visual Impulses

Anterior Thalamic Nucleus

Cerebral Cortex

Ascending Sensory Tracts
Reflex
Reflect

Layered Reflex

sense

move

Spine
Reflect

Layered Reflex

sense

move

Spine

Control Loop
Feed-Back Differential

Descending Neural Radiations
to the Hippocampus/
Hypothalamus

Corpus Callosum
Thalamus
Pineal Gland
Hippocampus
Cerebellum

Auditory Impulses
Projection to Spinal Cord
Ascending Tracts

of Hypothalamus
Amygdaloid Nucleus

Ascending Tracts

Layered architectures (cartoon)

- Cortex
- Neurons
- Cells

Physiology
Organs
Cells
Visual Cortex

Why?

Visual Thalamus

10x
What are the consequences?

Why?

There are 10x feedback neurons

Prediction
Goals

Conscious perception

Errors

Actions

Goals

Actions

Visual Cortex

Visual Thalamus

Conscious perception

10x

Why?

There are 10x feedback neurons

Prediction
Goals

Conscious perception

Errors

Actions

Goals

Actions

Visual Cortex

Visual Thalamus

Conscious perception

10x
Seeing is *dreaming*

Conscious perception

3D + time
Simulation

Conscious perception

Zzzzzz……
Same size?
Same size
Same size
Same size

Toggle between this slide and the ones before and after

Even when you “know” they are the same, they appear different
Same size?

Vision: evolved for complex simulation and control, not 2d static pictures

Even when you “know” they are the same, they appear different
3D Simulation + time + complex models ("priors")

Conscious perception

Seeing is *dreaming*

Zzzzzz…….
Seeing is \textit{dreaming}

Conscious perception

3D + time
Simulation + complex models ("priors")

Prediction

Conscious perception errors
Inferring shape from shading
Which blue line is longer?
Which blue line is longer?
Which blue line is longer?
Which blue line is longer?
Which blue line is longer?
Which blue line is longer?
Which blue line is longer?

With social pressure, this one.

Standard social psychology experiment.
Chess experts
• can reconstruct entire chessboard with < ~ 5s inspection
• can recognize 1e5 distinct patterns
• can play multiple games blindfolded and simultaneous
• are no better on random boards

(Simon and Gilmartin, de Groot)
Specialized Face Learning Is Associated with Individual Recognition in Paper Wasps

Michael J. Sheehan* and Elizabeth A. Tibbetts

We demonstrate that the evolution of facial recognition in wasps is associated with specialized face-learning abilities. *Polistes fuscatus* can differentiate among normal wasp face images more rapidly and accurately than nonface images or manipulated faces. A close relative lacking facial recognition, *Polistes metricus*, however, lacks specialized face learning. Similar specializations for face learning are found in primates and other mammals, although *P. fuscatus* represents an independent evolution of specialization. Convergence toward face specialization in distant taxa as well as divergence among closely related taxa with different recognition behavior suggests that specialized cognition is surprisingly labile and may be adaptively shaped by species-specific selective pressures such as face recognition.

When needed, even wasps can do it.
• *Polistes fuscatus* can differentiate among normal wasp face images more rapidly and accurately than nonface images or manipulated faces.
• *Polistes metricus* is a close relative lacking facial recognition and specialized face learning.
• Similar specializations for face learning are found in primates and other mammals, although *P. fuscatus* represents an independent evolution of specialization.
• Convergence toward face specialization in distant taxa as well as divergence among closely related taxa with different recognition behavior suggests that specialized cognition is surprisingly labile and may be adaptively shaped by species-specific selective pressures such as face recognition.
Fig. 1 Images used for training wasps.

P. fuscatus faces

Antenna-less faces

Rearranged faces

Patterns

Caterpillars

P. metricus faces

M J Sheehan, E A Tibbetts Science 2011;334:1272-1275

Published by AAAS
Unfortunately, we’re not sure how this all works.
Flexible/Adaptable/Evolvable

Horizontal Meme Transfer

Sensory

Prefrontal

Striatum

Learning

Reflex

Catabolism

AA

RN

AA trans

xRNA trans

DNA Repl

Gene

Software

Hardware

Horizonal App Transfer

Horizontal Gene Transfer

Digital

Analog

Depends crucially on layered architecture