Dynamical Systems: a Navigation Guide

Herbert Jaeger

university of groningen

Professor of Computing in Cognitive
Materials

CogniGron - Groningen Cognitive Systems
and Materials Center,
University of Groningen, NL

Mission (both of CogniGron and
CogniHerbert):

*Develop materials-centred systems
paradigms for cognitive computing based on
modelling and learning at all levels: from
materials that can learn to devices, circuits
and algorithms*

My background: AI, theoretical CS, machine
learning, neural networks, dynamical systems

What you might expect and what you will get

My guess about your expectations

- you have heard about "attractors", "bifurcations", "chaos" ("ABC")
- you have seen that ABC concepts are often applied to neural / cognitive dynamics
- you want to learn more about ABC

ABC is powerful and beautiful and insightful.

ABC is limited. There is also DEF ... XYZ.

What I try to give you

- a glimpse of the astonishing variety of concepts and tools available for dynamical modeling: AB...YZ
- a compass to navigate
- and yes, a special consideration of ABC

Overview

1. So many views on „dynamics"
2. A zoo of finite-state models
3. A ménagerie of continuous-state models
4. What is a state? ... and Takens' theorem
5. State-free modeling of temporal systems
6. Qualitative theory of DS: attractors, bifurcations, chaos
7. Non-autonomous dynamical systems

Slides (including references) online at
https://www.ai.rug.nl/minds/uploads/IntroDynSys_IK_2021.pdf

1. So Many Views on Dynamics

Literature

If you have about 6000 Euros to spare, the Encyclopedia of Complexity and Systems Science (R. A. Meyers, ed.), Springer Verlag 2008, is the definite compilation of dynamical systems knowledge (~600 detailed articles, > 10K pages, 11 volumes).

A bit more on the affordable side, at https://dsweb.siam.org/Education
you find an extensive compilation of online tutorials and course materials spanning the whole range of dynamical systems.
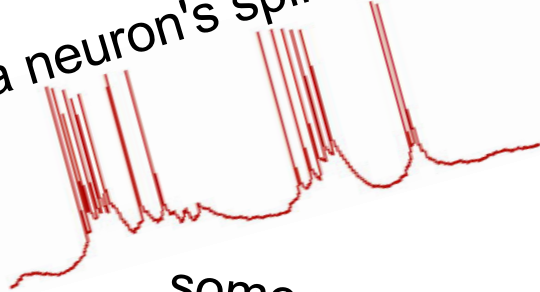
What is a dynamical system?

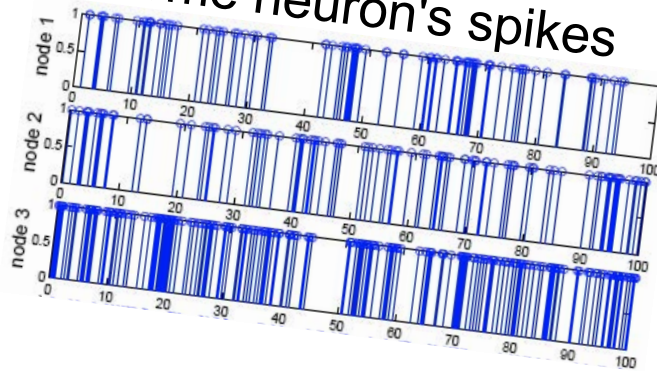A DS is any real or artificial or formal system that evolves over time.

It is almost impossible not to be a dynamical system!
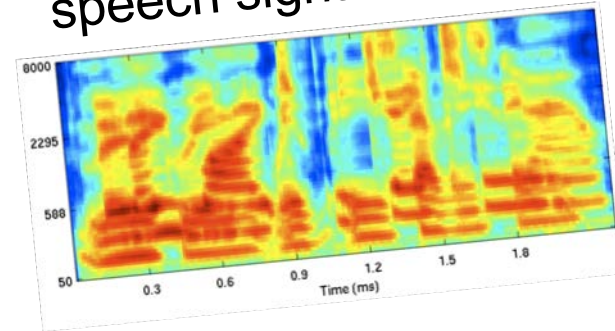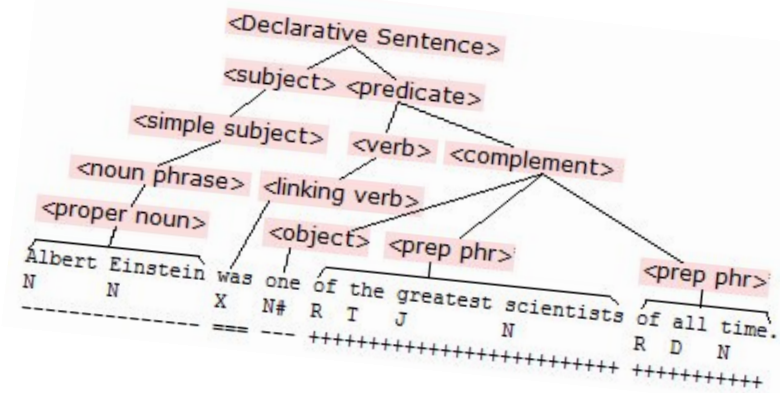
Things to describe & analyze

a neuron's spikes

speech signals

some neuron's spikes

language

<Declarative Sentence>

<subject> <predicate>

<simple subject>   <verb> <complement>

<noun phrase> <linking verb>

<proper noun>   <object> <prep phr>   <prep phr>

Albert Einstein was one of the greatest scientists of all time.
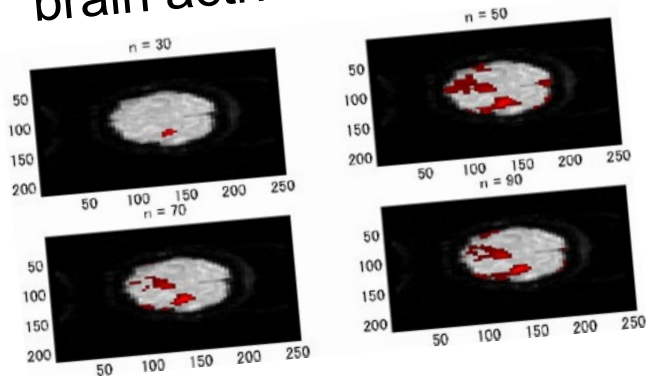
brain activation waves

controlled action

Examples

- A water molecule, a waterdrop, a river, an ocean
- The Universe
- A calcium channel, a synapse, a dendrite, a neuron, a microcircuit, ... a brain, a nervous system, a body
- Life on earth
- A bitstream, a network of communicating signal sources, a language generating program, a society of linguistic agents
- Mathematics (as a growing body of theorems and proofs)
- You
- A stone
- What you think about you

In sum:

- Everything that is not dead or boring.

There isn't and there can't be a universal theory of dynamical systems. We have to face a diversity of methods.

Types of dynamical systems and/or modeling methods

Two fundamental decisions before modeling starts:

- **selection**: what subsystem is modeled
- **perspective**: what aspects of that subsystem are modeled

These decisions mandate the use of very different modeling tools.

Symbolic ⟷ Numerical

Symbolic
Texts, event and action sequences, DNA, conceptual reasoning

Numerical
Physiological models, psychometrical measurements, motor control

Deterministic
Electrodynamics, artificial neural networks, mean-field models

Non-deterministic
Language competence models, grammatical sequence generation

Stochastic
spike trains, speech, language performance models

Autonomous
Sleep dynamics (?), central pattern generator models (?), circadian clocks (?)

Non-autonomous
well, ...almost every real-life system

... continued

low-dimensional ⟷ **high-dimensional**

Hodgkin-Huxley or FitzHugh-Nagumo model of neurons, oscillator models

network-level modeling, modeling of cognitive processes

discrete time ⟷ **continuous time**

state-switching models, models learnt from sampled data

classical neuron models, mean-field models of collective dynamics

linear ⟷ **non-linear**

"classical" analysis of neural dynamics as signals

neural pattern generators, chaotic dynamics, coupled oscillators

homogeneous ⟷ **non-homogeneous**

fully or sparsely connected neural network

modular or hierarchical neural circuits and architectures

non-stationary

learning processes, speech (short timescale), adaptation processes

stationary

neural noise, speech (long timescale), fruit fly in Andrew Straw's virtual arena

evolutionary

language evolution, ontogenesis, cell differentiation

Three modeling attitudes

Analytical modeling

- Try to capture the real-world underlying mechanisms
- The physicist's and neuro-medicinal view
- Examples: compartment models, Hodgkin–Huxley model, Chomsky generative grammar
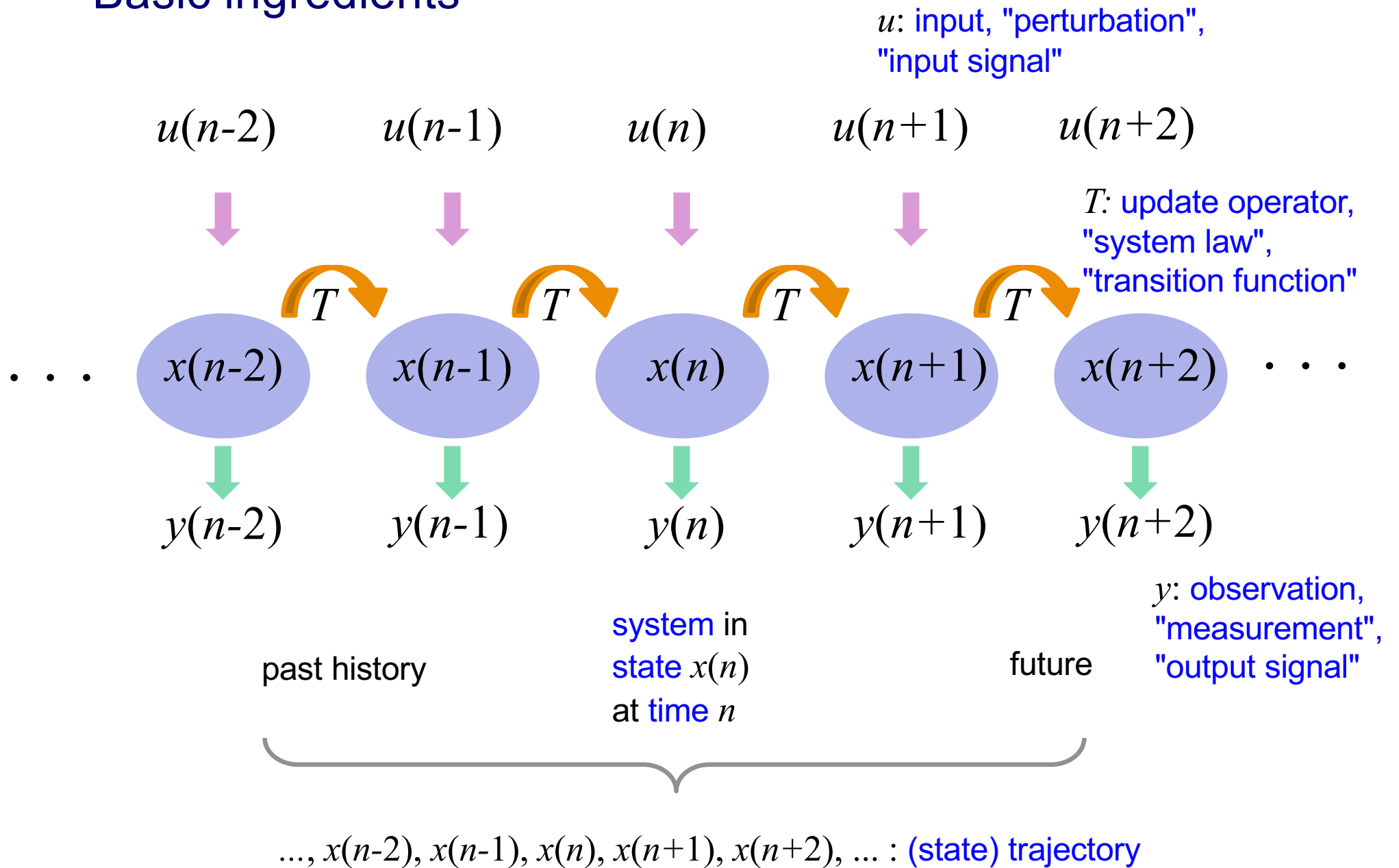- Formalisms: ODEs, PDEs, automata models, dynamical Graphical Models

Blackbox modeling

- Try to describe the observable phenomena
- The approach of machine learning, signal processing
- Examples: BCI applications, brain data analyses, artificial intelligence systems
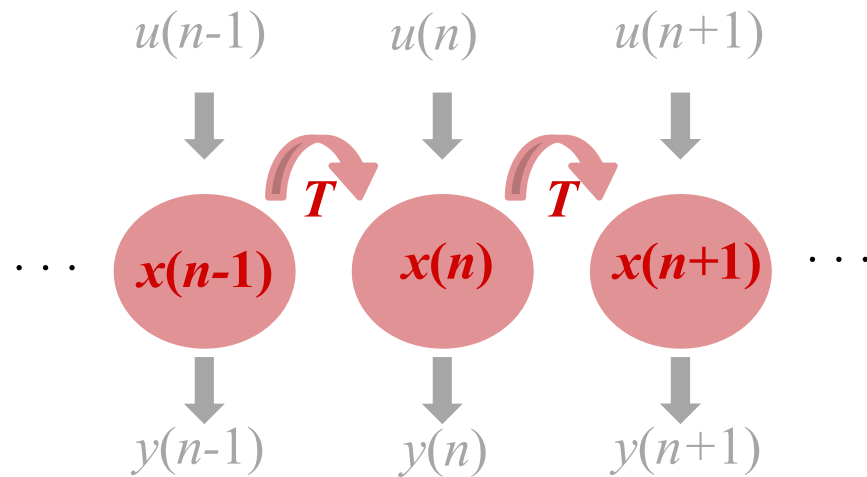- Formalisms: hidden Markov models, stochastic DEs, artificial neural networks

Epistemic modeling (rare)

- Try to emulate information gain processes
- Sometimes adopted by roboticists, agent modeling
- Examples: concept formation processes, belief state modeling
- Formalisms: predictive state representations, observable operator models, temporal logics
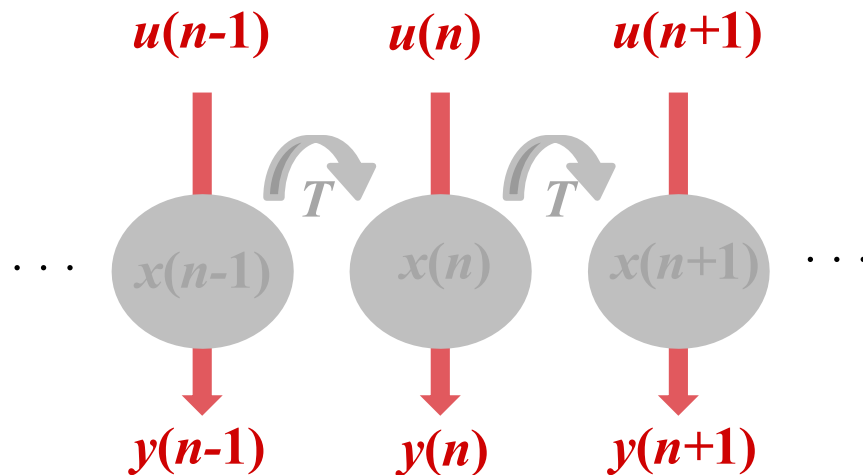
Basic ingredients

u: input, "perturbation", "input signal"

$u(n\text{-}2)$ $u(n\text{-}1)$ $u(n)$ $u(n\text{+}1)$ $u(n\text{+}2)$

T: update operator, "system law", "transition function"

$\cdots$ $x(n\text{-}2)$ T $x(n\text{-}1)$ T $x(n)$ T $x(n\text{+}1)$ T $x(n\text{+}2)$ $\cdots$

$y(n\text{-}2)$ $y(n\text{-}1)$ $y(n)$ $y(n\text{+}1)$ $y(n\text{+}2)$

y: observation, "measurement", "output signal"

past history

system in state $x(n)$ at time n

future

$..., x(n\text{-}2), x(n\text{-}1), x(n), x(n\text{+}1), x(n\text{+}2), ...$: (state) trajectory
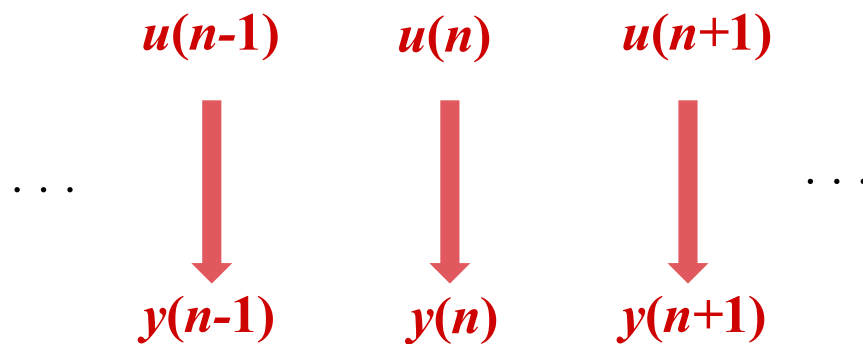
The natural science view

- ever since Newton

- objective: understand "the system"

- focus on modeling the system state x and the system law T

- classical formalism: ordinary differential equations (ODEs) for T (continuous time), $x \in R^n$

- classical approach: isolate system in experimental designs, minimizing role of perturbations u, making experiments reproducible

- that is, try to ensure that system can be treated as an autonomous system

- main role of output y: measurable, that is, a vehicle to infer back to state x

The engineering view

$u(n-1)$ $u(n)$ $u(n+1)$

$\cdots$ $x(n-1)$ $x(n)$ $x(n+1)$ $\cdots$

$y(n-1)$ $y(n)$ $y(n+1)$

"state-based representation"

$u(n-1)$ $u(n)$ $u(n+1)$

$\cdots$ $\cdots$

$y(n-1)$ $y(n)$ $y(n+1)$

that's what a classically trained engineer calls a "system", or "filter"

- the classical perspective taken by signal processing and control engineers

- objective: design and build useful input-output devices ("filters", "controllers")

- focus on the mathematical relationships between input signals $u(n)$ and output signals $y(n)$

- engineers love linear input-output relationships: highly developed arsenal of mathematical methods ("frequency domain" modeling)

- "state-based representations" are only one (and not the classical) of the modeling approaches in the signal processing field.

Literature

There are many textbooks on signal processing. Free online textbooks that try to be intuitive and do not require much maths are listed at the end of http://en.wikipedia.org/wiki/Signal_processing. A more rigorous yet accessible and comprehensive textbook treatment is provided by the online lecture notes of the MIT open course

Alan Oppenheim, and George Verghese. *6.011 Introduction to Communication, Control, and Signal Processing, Spring 2010*. (Massachusetts Institute of Technology: MIT OpenCourseWare), http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-011-introduction-to-communication-control-and-signal-processing-spring-2010/

A first comparison
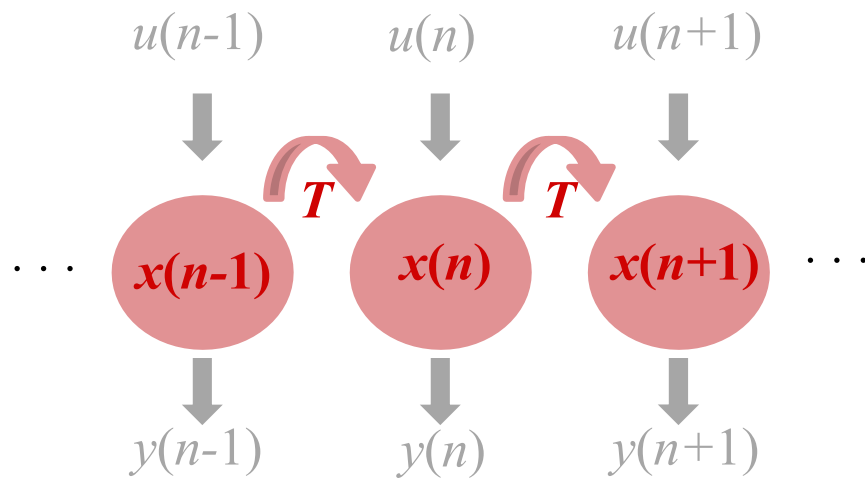
- physicists / neurophysiologists / neurologists / neurolinguists want to model neural architectures / mechanisms

- tools from dynamical systems theory proper, "ABC" tradition, stochastic processes and information theory
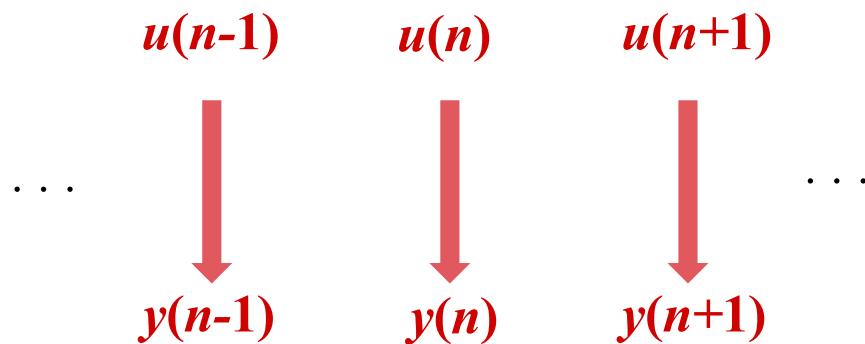
- cognitive scientists / linguists / BCI engineers / machine learners want to model neural function / performance

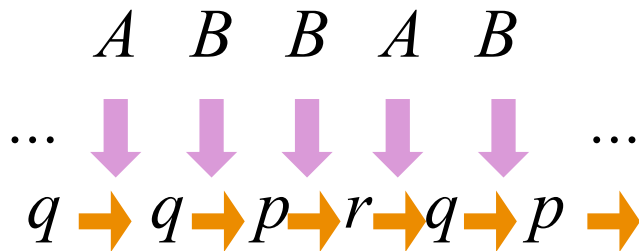- tools from signal processing, machine learning, CS, stochastic processes and information theory

It's good to be familiar with *both*.

2. A Zoo of Finite-State Models

Deterministic finite-state automata (DFA)
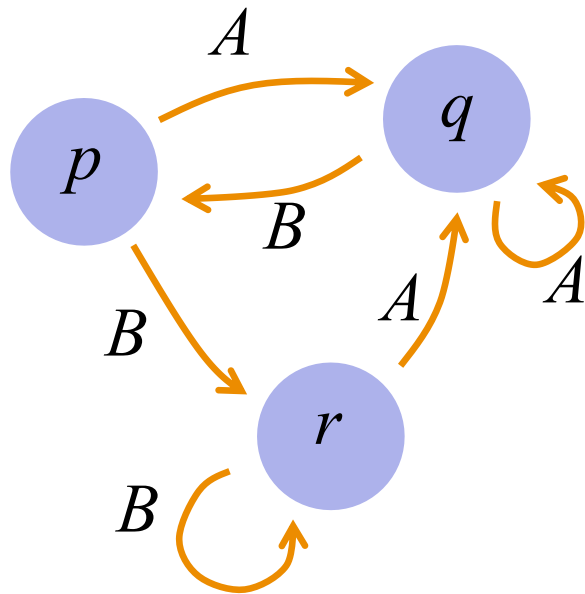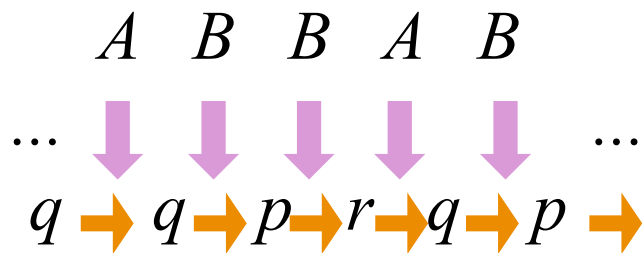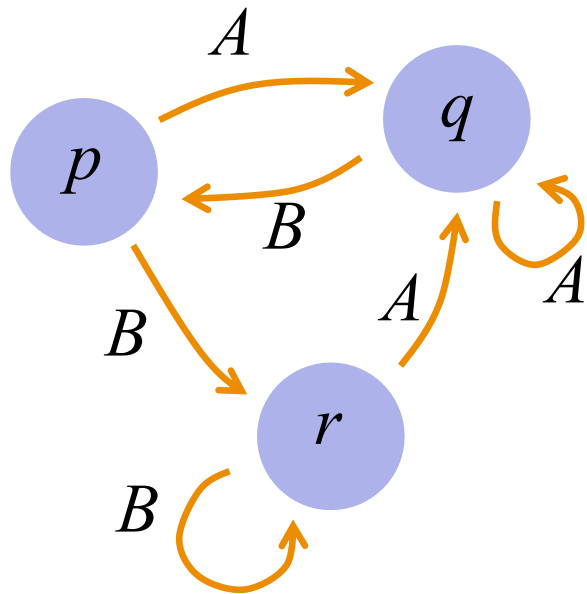

example trajectory:

A DFA is defined by:

- a finite set of states, e.g. $Q = \{p, q, r\}$

- a finite set of input symbols, e.g. $\Sigma = \{A, B\}$

- a transition function $T: Q \times \Sigma \to Q$, can be written as table, e.g.

|	A	B
p	q	r
q	q	p
r	q	r

- A DFA defines input-sequence dependent state trajectories

DFAs, comments

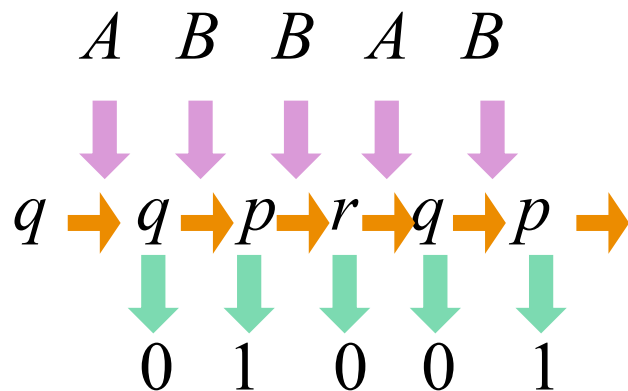- deterministic

- can be used e.g. for modeling ion channel states or an agent's model of how the world Q reacts on agent's actions Σ

- simplicity is deceptive: a PC can be considered a DFA, with more-than-astronomical-sized (but finite)
  $Q = \{$`all possible logic gate state combinations`$\}$

- states are "fully observable"

- inferring a DFA from observed trajectories is easy

- DFAs are standard tool for theoretical CS, then used only for finite sequences ("words")

Moore- and Mealy-Machines
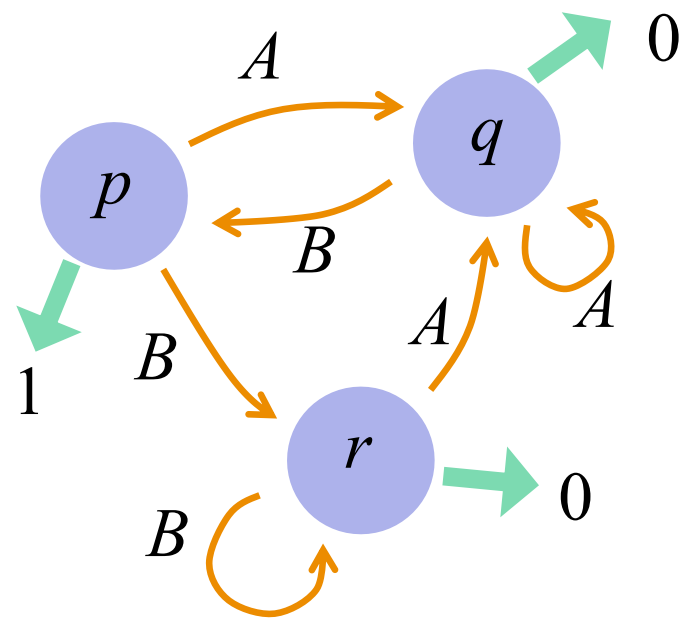
example Moore trajectory:

A Moore Machine is a DFA, equipped additionally with

- a finite set of output symbols, e.g. $O = \{0, 1\}$

- a translation (observation) function $\rho: Q \rightarrow O$

A Mealy Machine is similar, except the observations are "emitted" from transition arrows: $\rho: Q \times \Sigma \rightarrow O$

Both are deterministic.

Efficient methods to infer Moore / Mealy machines from input-output data are known.

Literature

There are many textbooks covering finite automata – they form a core part of theoretical CS and any theoretical CS textbook will cover them (among other topics). A classic is

Hopcroft, John E. , Motwani, Rajeev, and Ullman, Jeffrey: *Introduction to Automata Theory*, 2nd edition. Addison-Wesley, 2001

Textbook concentrating on finite automata which covers both the use of automata for finite-word languages and for infinite-sequence languages

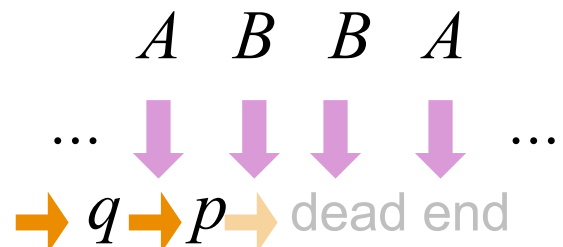A. de Vries: Finite Automata: Behavior and Synthesis. Elsevier, 2014

Non-deterministic finite-state automata (NFA)

A NFA is defined by:

- a finite set of states, e.g. $Q = \{p, q, r\}$

- a finite set of input symbols, e.g. $\Sigma = \{A, B\}$

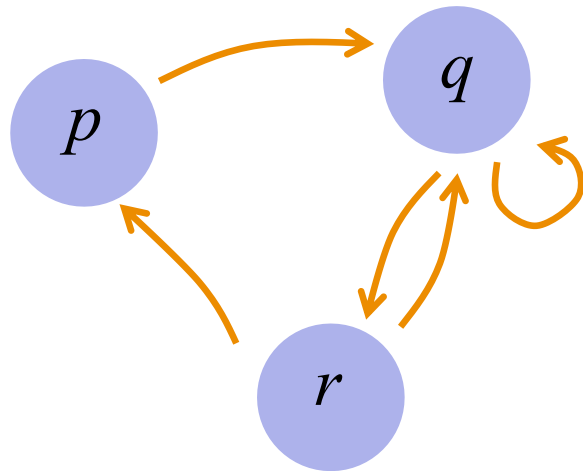- a transition function $T: Q \times \Sigma \rightarrow \mathrm{Pot}(Q)$, (Pot: power set), e.g.
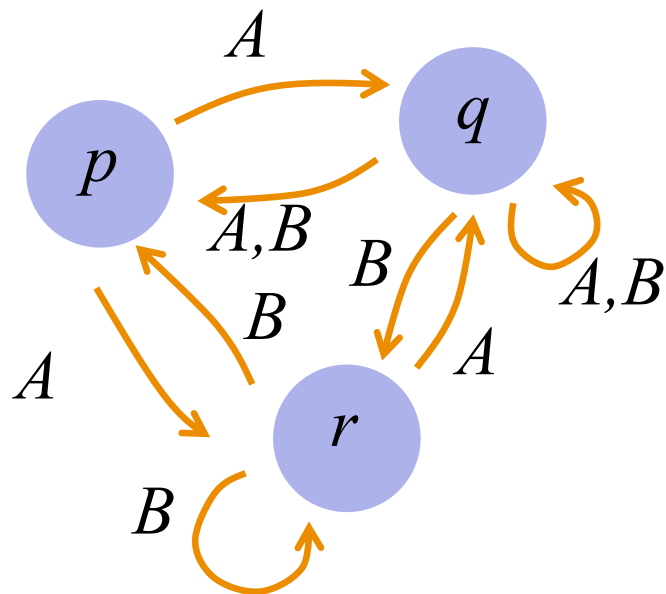
example trajectories:

$A \quad B \quad B \quad A$

... ⬇ ⬇ ⬇ ⬇ ...

→ q → p → dead end

or → p → r → r → p → q →

	A	B
p	$\{q, r\}$	$\emptyset$
q	$\{p, q\}$	$\{p, q, r\}$
r	$\{q\}$	$\{p, r\}$

NFAs, comments

special case: no input
symbols

example trajectories:

$q \Rightarrow q \Rightarrow r \Rightarrow p \Rightarrow q \Rightarrow r \Rightarrow$
$q \Rightarrow q \Rightarrow q \Rightarrow q \Rightarrow r \Rightarrow q \Rightarrow$

- an NFA can yield several (typically, infinitely many) different state sequences on given input sequence

- no probabilities involved; a given state sequence cannot be said to be "more probable" than another

- this is called a non-deterministic system, as opposed to "deterministic" and to "stochastic"

- nondeterministic models capture what is possible vs. what is impossible to observe

- special case: no input (or equivalently, one-element input set)

Finite-dimensional Markov chains
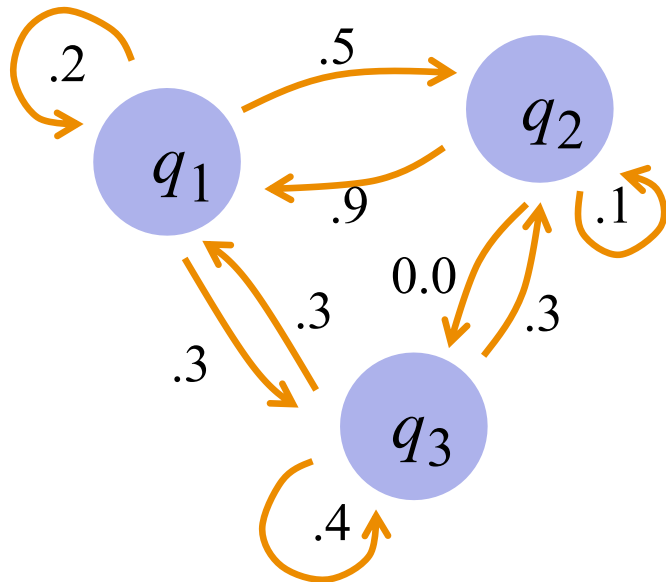
A (finite-dimensional) Markov chain (MC) is defined by:

- a finite set of states, e.g. $Q = \{q_1, q_2, q_3\}$

- an initial state distribution $\boldsymbol{p} \in Prob(Q)$, where $Prob(Q)$ is the set of probability distributions over Q

- a transition kernel $T: Q \rightarrow Prob(Q)$

- T can be written as stochastic transition matrix ("Markov matrix"), e.g.

$$\boldsymbol{p} = \begin{pmatrix} 0.5 \\ 0.0 \\ 0.5 \end{pmatrix}$$

	q_1	q_2	q_3
q_1	0.2	0.5	0.3
q_2	0.9	0.1	0.0
q_3	0.3	0.3	0.4

rows sum to 1

Finite-dimensional MCs, comments 1
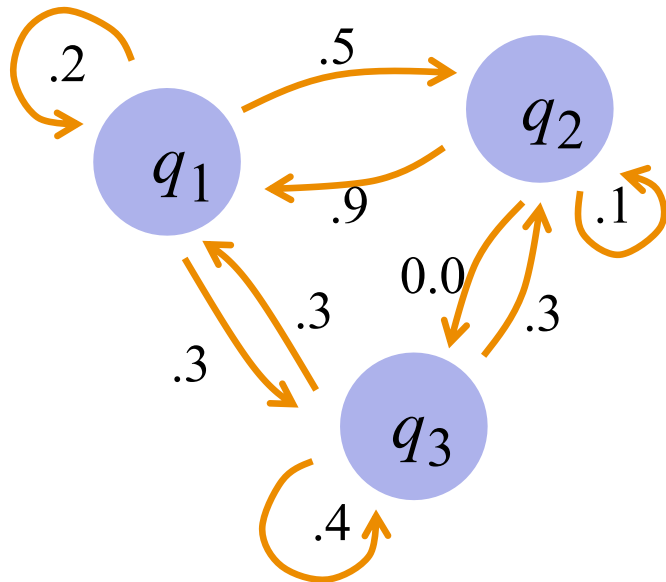
$$p = \begin{pmatrix} 0.5 \\ 0.0 \\ 0.5 \end{pmatrix}$$

- used to describe trajectories that start at time $n = 0$

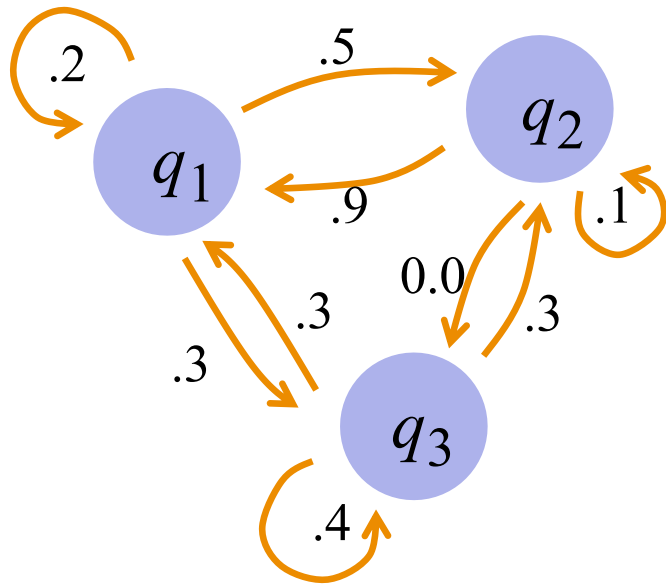- probability that a trajectory starts with

$$q_{i0}, q_{i1}, ..., q_{in}$$

is

$$P(X_0 = q_{i_0}, ..., X_n = q_{i_n}) =$$

$$= P(X_0 = q_{i_0}) \cdot P(X_1 = q_{i_1} \mid X_0 = q_{i_0}) \cdot ... \cdot$$

$$\cdot P(X_n = q_{i_n} \mid X_{n-1} = q_{i_{n-1}})$$

$$= P(X_0 = q_{i_0}) \cdot \prod_{k=1}^{n} P(X_k = q_{i_k} \mid X_{k-1} = q_{i_{k-1}})$$

$$= p(i_0) \, T(i_0, i_1) ... T(i_{n-1}, i_n)$$

- a MC specifies a stochastic process

$$(X_n)_{n=0,1,2,...}$$

with values in Q

Finite-dimensional MCs, comments 2

$$p = \begin{pmatrix} 0.5 \\ 0.0 \\ 0.5 \end{pmatrix}$$

- the crucial defining property to make a finite-valued stochastic process a Markov chain: the Markov property

$$P(X_{n+1} = q_{i_{n+1}} \mid X_0 = q_{i_0}, ..., X_n = q_{i_n}) =$$
$$= P(X_{n+1} = q_{i_{n+1}} \mid X_n = q_{i_n})$$

- What is going to happen next (probabilities to observe q_{in+1}) only depends on current state q_{in}, not on previous state history
- MCs are "memoryless" systems

Controlled Markov chains
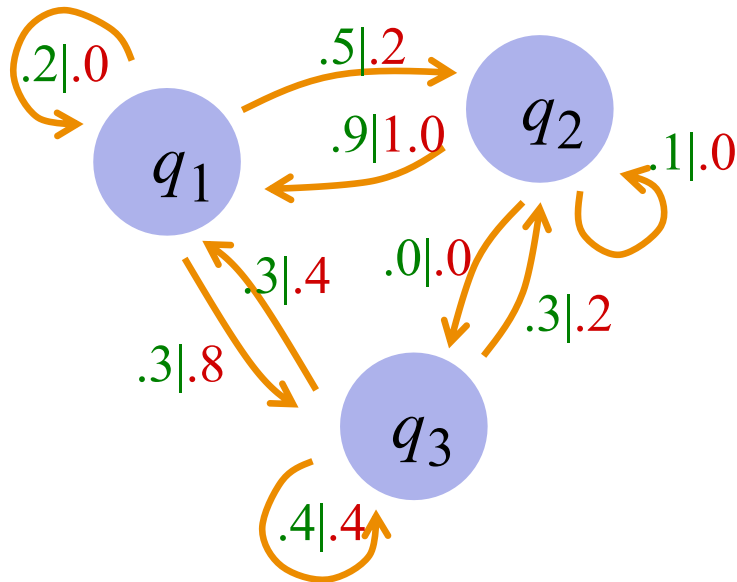
In a controlled MC the transition probabilities are switched by inputs. Components:

- a finite set of states, e.g. $Q = \{q_1, q_2, q_3\}$

- a finite set of inputs ("control actions"), e.g. $A = \{a, b\}$

- an initial state distribution $p \in Prob(Q)$

- for each $a \in A$, a transition kernel $T_a: Q \rightarrow Prob(Q)$

Diagram labels:

q_1, q_2, q_3

.2|.0, .5|.2, .9|1.0, .1|.0, .3|.4, .0|.0, .3|.2, .3|.8, .4|.4

$$p = \begin{pmatrix} 0.5 \\ 0.0 \\ 0.5 \end{pmatrix}$$

T_a	q_1	q_2	q_3
q_1	0.2	0.5	0.3
q_2	0.9	0.1	0.0
q_3	0.3	0.3	0.4

T_b	q_1	q_2	q_3
q_1	0.0	0.2	0.8
q_2	1.0	0.0	0.0
q_3	0.4	0.2	0.4

- Update mechanism: switch transition kernel according to current input symbol

Literature

A classical monograph on controlled stochastic processes (general rigorous mathematical theory, not restricted to controlled MCs):

Gihman, I.I. and Skorohod, A.V., Controlled Stochastic Processes. Springer Verlag 1979

Controlling stochastic systems is, of course, also of prime importance in control engineering. In this field, the controlled systems often are systems that emit observable output, which is then included in the analysis and methods. A textbook:

R. F. Stengel, Stochastic optimal control: theory and application. John Wiley and Sons, 1986

Hidden Markov models (HMMs)

example trajectory:

→ *q* → *p* → *r* → *q* → *p* →

0 1 0 0 1
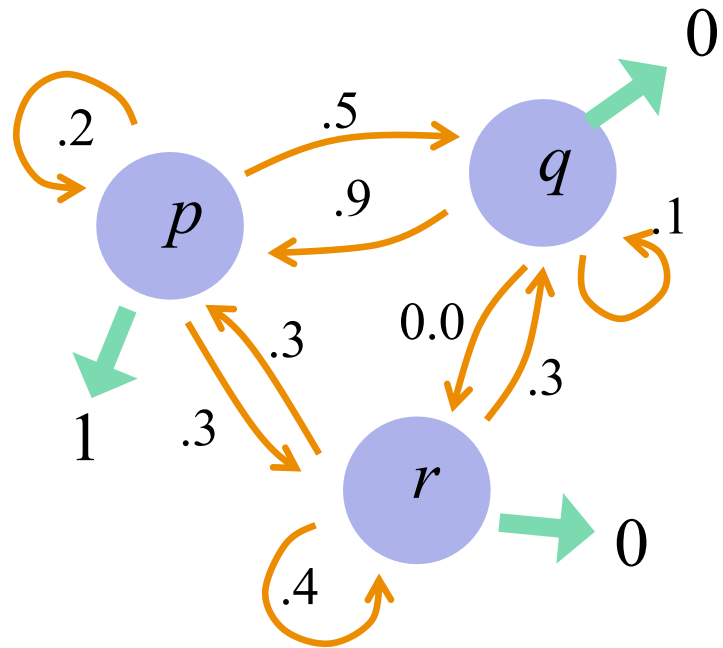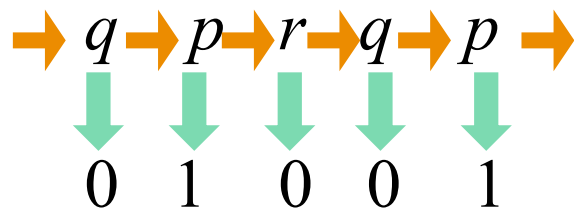
Defining components:

- a finite set of states, e.g. $Q = \{p, q, r\}$
- a finite set of outputs ("observables", "visibles"), e.g. $O = \{0, 1\}$
- an initial state distribution $\boldsymbol{p} \in Prob(Q)$
- a transition kernel $T : Q \to Prob(Q)$
- for every state $q \in Q$ and observable $o \in O$, an emission probability $P(o|q)$ to observe o when the hidden Markov state trajectory passes through q
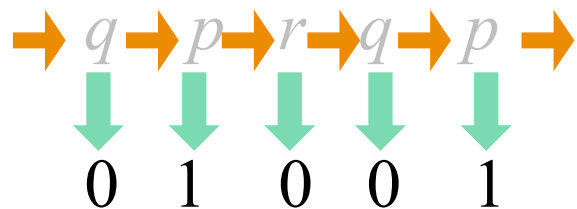
  or, equivalently,

an emission function $\rho : Q \to O$ (as shown in example)

HMMs, comments 1
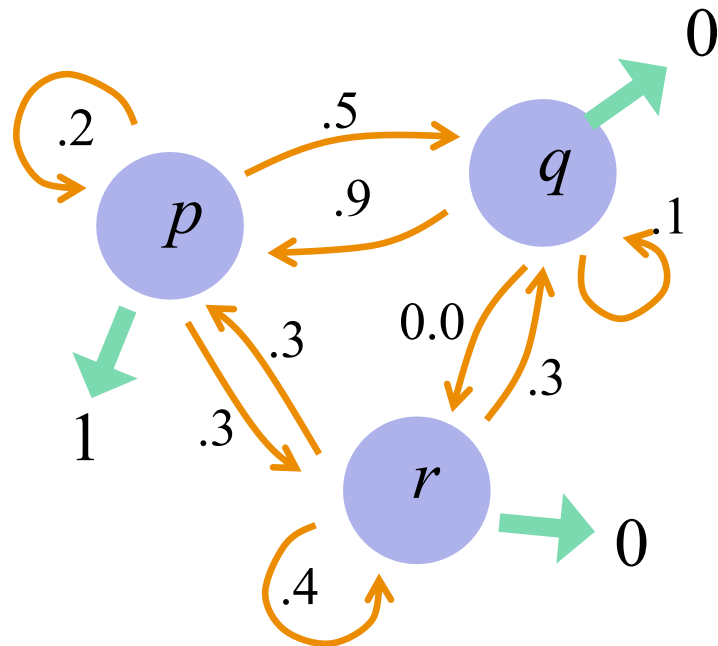
unobservable states

$\rightarrow q \rightarrow p \rightarrow r \rightarrow q \rightarrow p \rightarrow$

0 1 0 0 1

visible measurements

- widely and naturally applicable, because an experimenter often can't directly observe states q, only make measurements o of them

- available experimental data are only trajectories of observables, states are unobservable

- model inference task: from (empirical) measurement data (e.g., 0 1 0 0 1) infer underlying stochastic state transition system, that is...

- ... explain data by generative mechanism

- Example: Q = "brain states", O = "uttered phonemes"

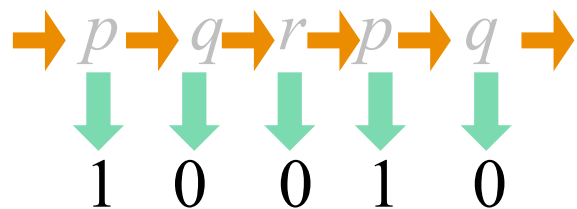- Example: Q = "state of a neuron", O = "spike"

HMMs, comments 2
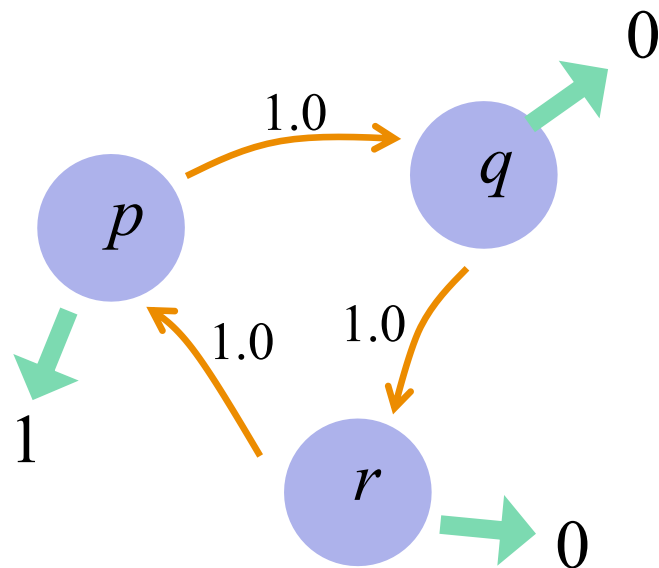
- The visible trajectories (values of random variables Y_n) "have memory":

$$P(Y_{n+1} = 1 \mid Y_n = 0) = 0.5$$
$$P(Y_{n+1} = 1 \mid Y_n = 0, Y_{n-1} = 0) = 1.0$$

- The observables $(Y_n)_{n=0,1,2,...}$ form a stochastic process in their own right, but this process does not have the Markov property

hidden states X_n

visible measurements Y_n

Literature

The classical tutorial text on HMMs, very readable (30000 Google cites, boosted the popularity of HMMs in speech processing):

Rabiner, L., A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77.2 (1989): 257-286. (many online copies)

Controlled hidden Markov models, aka POMDPs
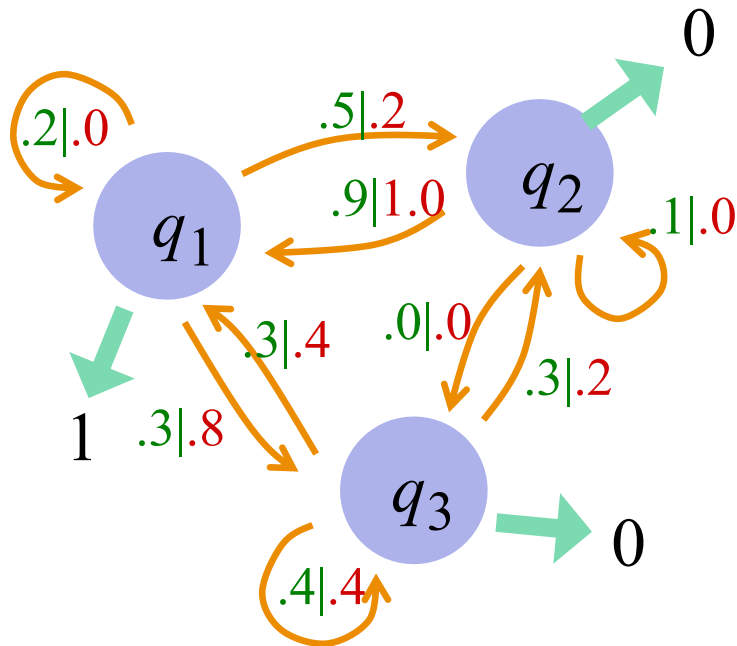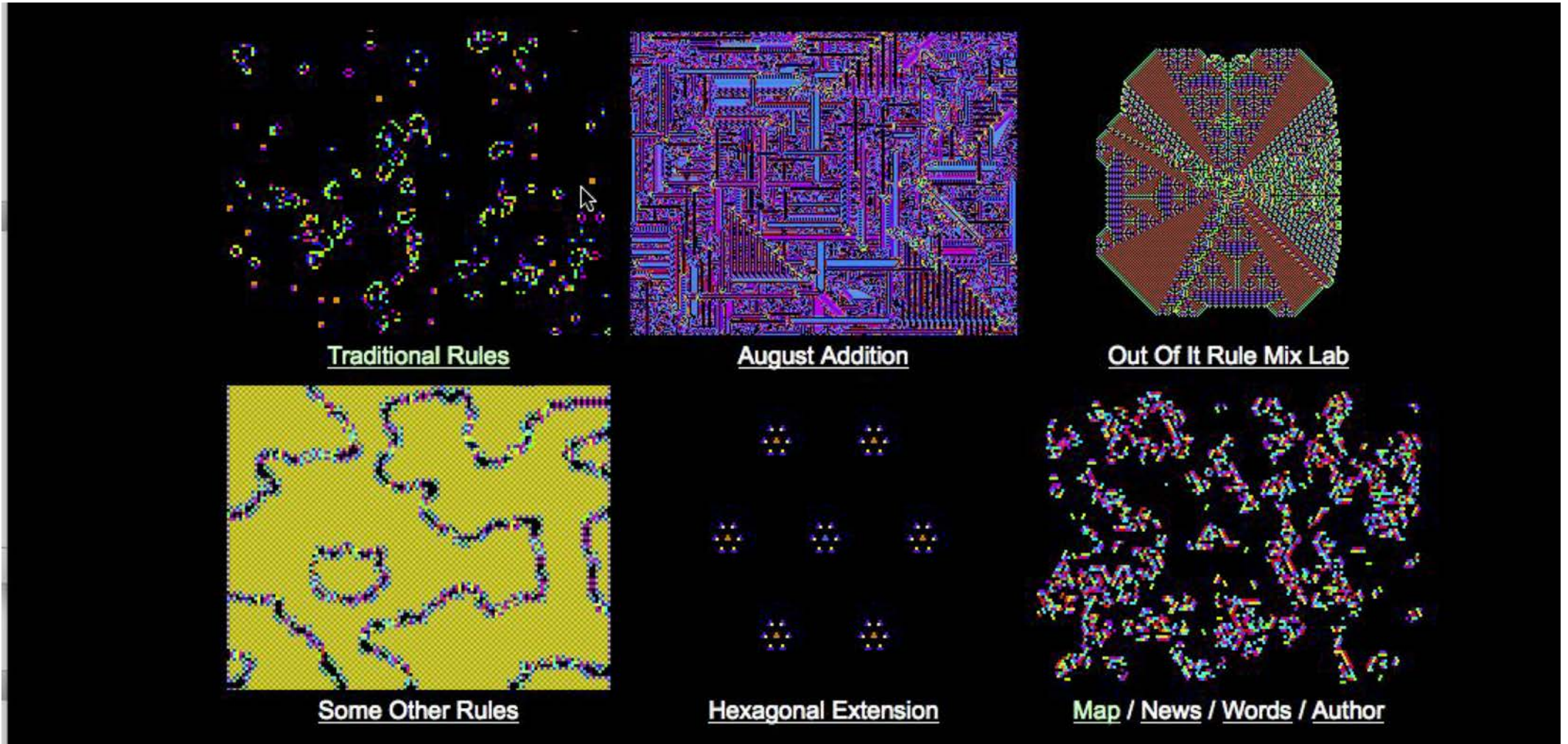

- crossover of controlled MCs and HMMs
- also (widely) known as Partially Observable Markov Decision Processes (POMDPs) if rewards and decision-making policy are added to the picture
- a basic tool in theory of autonomous agents / robotics / reinforcement learning in the machine learning sense
- In that context, a POMDP constitutes the agent's world model:
  - Q: external world states
  - A: agent's actions in world
  - O: sensory feedback from world
- methods available for learning a POMDP from A-O (action – sensor-feedback) timeseries data
- seems a natural model class to me also for neural dynamics and animal behavior

Literature

My favorite tutorial text on POMDPs, set in a context of agent learning and reinforcement learning:

Kaelbling, L.P., Littman, M.L., Cassandra, A.R., Planning and acting in partially observable stochastic domains. Artificial Intelligence 101 (1998), 99-134

Cellular automata (CA) – visual demo

Traditional Rules   August Addition   Out Of It Rule Mix Lab

Some Other Rules   Hexagonal Extension   Map / News / Words / Author

http://www.collidoscope.com/modernca/welcome.html (now defunct).
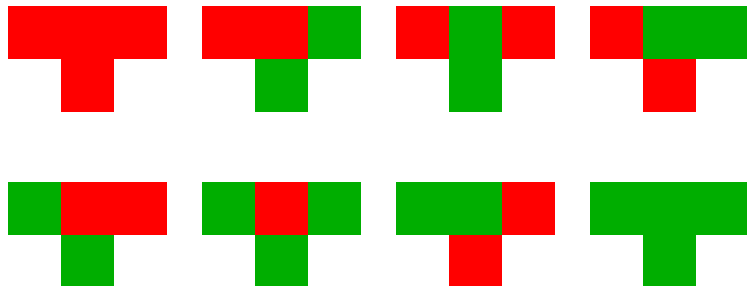A nice interactive CA simulator is at https://www.fourmilab.ch/cellab/

1-dim deterministic CA

Defining components:

- a finite set of local cell states, e.g.
  $Q = \{red,\ green\}$ (visualize as colors)

- a local transition function

$$T_{local}: Q \times Q \times Q \to Q$$
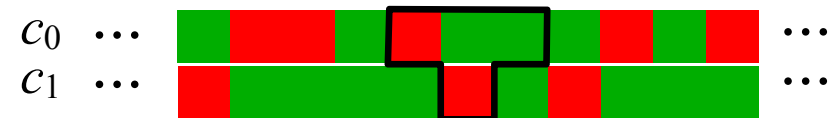
Example:

How it works:

1. Initial configuration: $c_0: \mathbb{Z} \to Q$

c_0 ... ...

2. Apply local transition function to all local neighborhoods (cell triplets), get next configuration c_1

c_0 ... ...
c_1 ...

3. Iterate, obtain trajectory of configurations $c_0, c_1, c_2,...$

A CA defines a global transition function T: {configurations} $\to$ {configurations}

CAs, comments

- can be defined for 1-dim, 2-dim, ... systems

- simplest kind of model for spatio-temporal dynamics

- local transition rules can be also defined stochastically $\tau: Q \times Q \times Q \to Prob(Q)$

- PDE models can be approximated by CAs via discretization

- Popular to model pattern formation

- Popular to analyze self-organization classes in spatiotemporal systems

- Suggested uses in neuroscience:
  - neural field models
  - classification of cortical dynamics
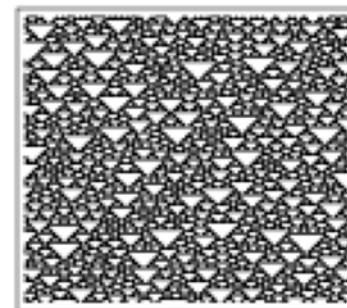 - computational power analysis of neural circuits

Fowler et al, 1992
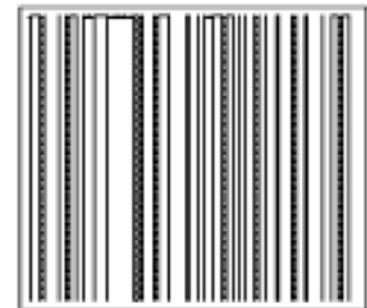
Wolfram Class 1   Class 2

Class 3   Class 4

Wolfram, 2002

Literature

The popularity of CAs is very much owed to Stephen Wolfram's lifelong passion about them and his missionary skills (paired with unlimited self-confidence). His most recent book (in a long series) is

Wolfram, S. A New Kind of Science. Wolfram Media, 2002. Online at http://www.wolframscience.com/nksonline/toc.html

A simply beautiful paper about seashell modeling, part of which is done with CA:

Fowler, D. R. and Meinhardt, H. and Prusinnkiewicz, P., Modeling seashells. In Proc. SIGGRAPH 92, (Computer Graphics 26, ACM SIGGRAPH) 1992, 379-387

Petri nets

- An application / engineering oriented formalism for modelling spatially distributed systems with flows and transformations of materials or information
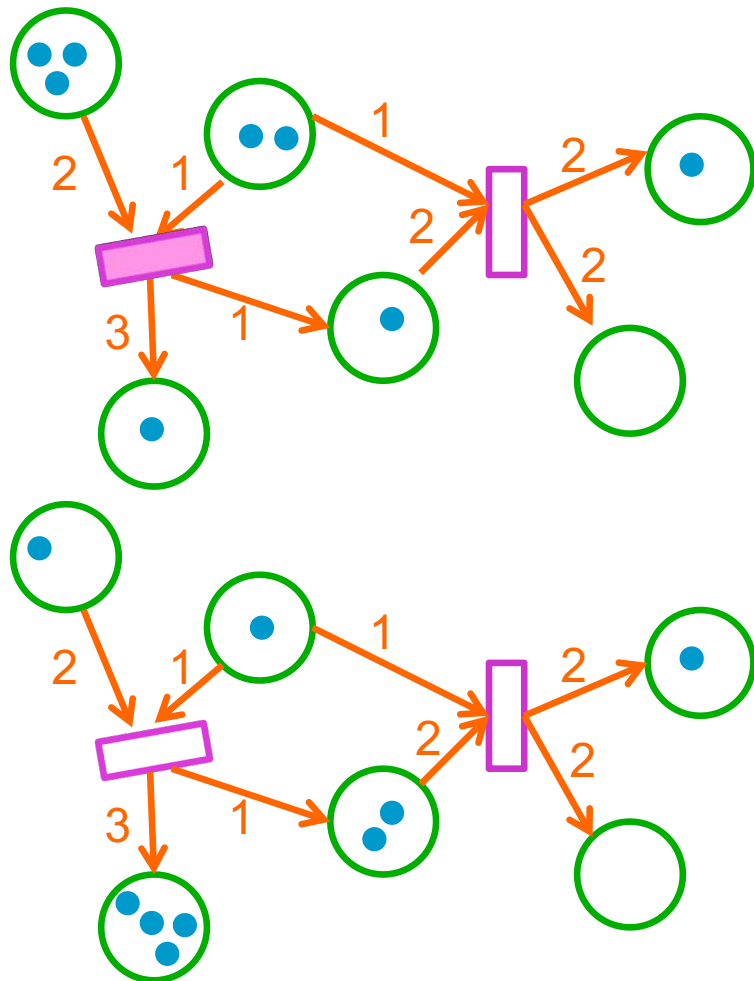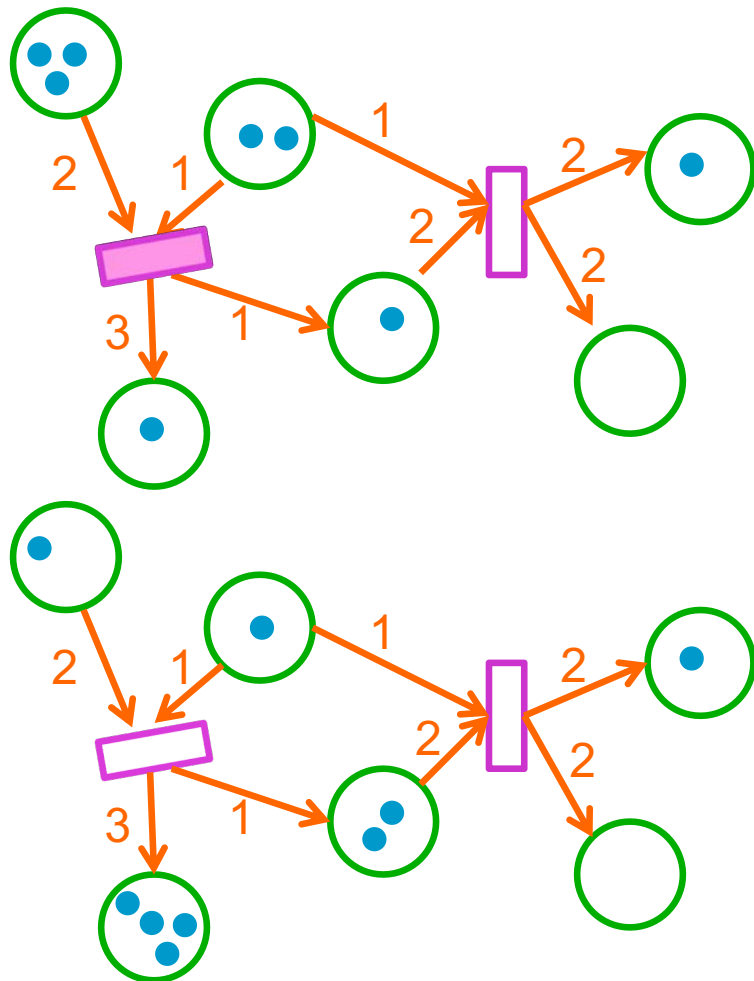

Components:

Places, transitions, arcs, tokens.

Configuration update operation:

1. Find some *enabled* transition

   (i.e., each "feeding" place has enough tokens for input arc weight)

2. Consume input tokens, create output tokens (in numbers given by output-arc weights)

Petri nets, comments

- Originated by early informatics researcher Carl Adam Petri in the 1960ies as formalism to capture *concurrency* of distributed, discrete-state switching processes

- Wide diversity of application domains, e.g.
  - asynchronous switching circuits,
  - parallel programming,
  - transportation and manufacturing logistics,
  - business processes

- Many variations / extensions of formalism

- Active community

- According to Wikipedia, Petri Nets offer a very useful compromise between expressiveness and analyzability of questions like:
  - is a target configuration reachable from an initial configuration?
  - does a process terminate?
  - which configurations are reversible to an initial / safe configuration?

Literature

I can recommend the English and German Wikipedia pages on "Petri Net" and "Petri Netz", respectively.

# Dynamical Bayesian networks, dynamical graphical models
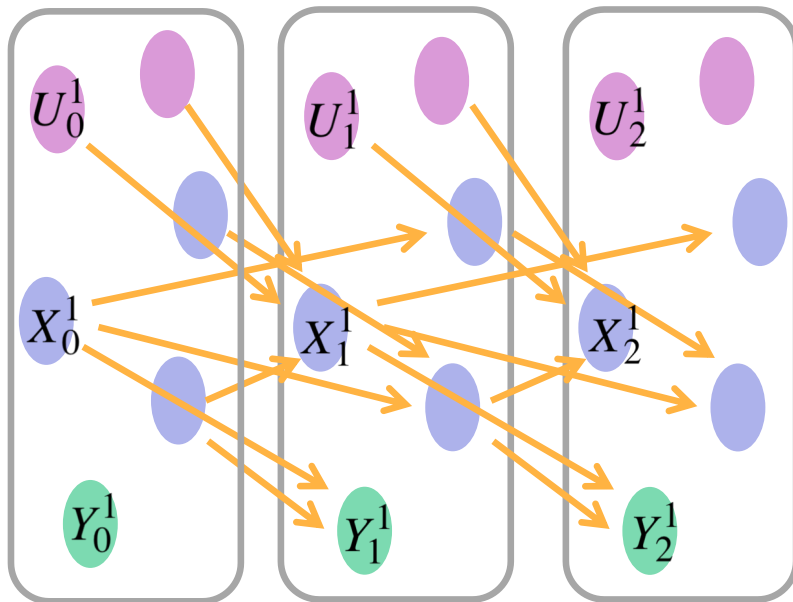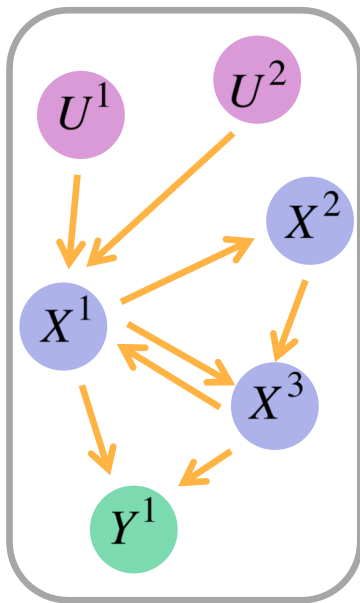
- Idea: describe the state of a stochastic DS at time n by the values of a finite collection of finite-valued
  - input random variables (RVs) U_n^i
  - hidden RVs X_n^j
  - output RVs Y_n^k
- These are causally interacting along statistical dependency chains leading from inputs through hiddens to outputs (there may be causal cycles between the hiddens)
- This dependency network is unfolded in time
- Local conditional probability distributions for hidden and output variables V

$$P(V_{n+1}^i \mid \text{ entire trajectory history} \leq n) =$$

$$P(V_{n+1}^i \mid \text{ values of parents of } V^i \text{ at time } n)$$

allow one to compute the distribution of the entire controlled stochastic process

Dynamical Bayesian networks, comments

U_0^1 U_1^1 U_2^1

X_0^1 X_1^1 X_2^1

Y_0^1 Y_1^1 Y_2^1

...
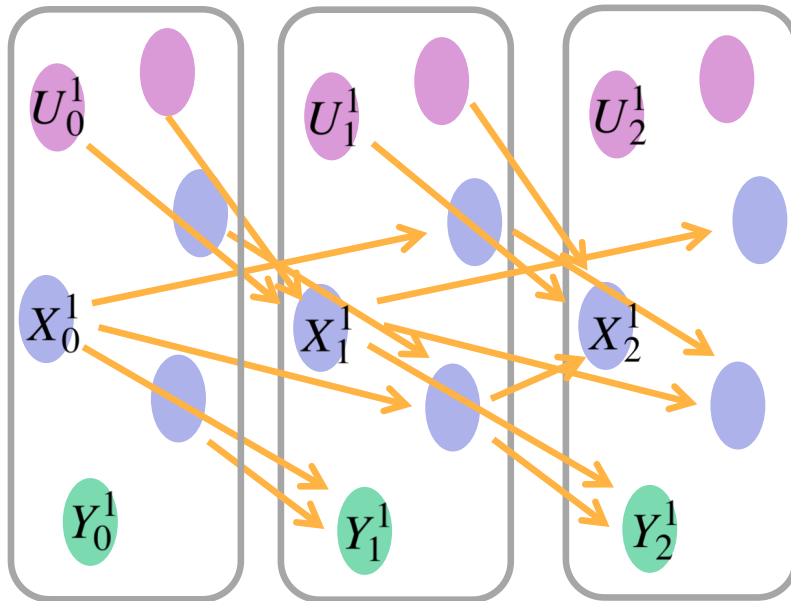
- DBNs can be seen as multivariate generalization of HMMs and POMDPs

- The natural model class for interacting stochastic quantities

- Also useful for analysis of multivariate empirical timeseries

- Extensive literature in machine learning, well-developed mathematical theory, algorithms available for

  - inference: prediction, filtering, data completion, optimal control

  - learning: given input-output data, estimate a DBN with hiddens that can re-generate the observed input-output relationships

- Exemplary suggested uses:

  - modeling functional connectivity networks

  - blackbox modeling of multivariate timeseries

  - high-level models of cognitive probabilistic reasoning

Literature

The highly cited, transparently written, career-making PhD thesis on dynamical Bayesian networks:

Murphy, Kevin Patrick. *Dynamic Bayesian networks: representation, inference and learning*. Diss. University of California, Berkeley, 2002. (many online pdf copies)

The ultimate handbook on graphical models in general:

Koller, D. and Friedman, N. Probabilistic Graphical Models. MIT Press 2009 (1200 pages!)

Finite state models, general comments

The zoo

- DFAs, NFAs

- finite-state MCs

- controlled MCs

- HMMs

- POMDPs

- cellular automata

- dynamical graphical models

- ...

- typically set-up in discrete time

- for all these models, "learning" algorithms are known ("model estimation")

- simpler to describe, analyze and, emphatically, to learn (= infer from empirical observation data) than continuous-time or continuous-valued models

- don't under-estimate their power and complexity: with growing number of states models can come as close as one wishes to continuous models

- lend themselves more easily to information-theoretic analyses than continuous models

3. A Ménagerie of Continuous-State Models

Iterated function systems, aka iterated maps

$$\mathbf{x}(n+1) = T(\mathbf{x}(n))$$

simplest case: autonomous system, i.e. no input, deterministic update

$$\mathbf{x}(n+1) = T(\mathbf{x}(n), \mathbf{u}(n)) + v$$

$$\mathbf{y}(n+1) = R(\mathbf{x}(n+1))$$

enriched with input $\mathbf{u}$ and additive noise v: non-autonomous system. Further addition: output function R.

$$\mathbf{x}(n+1) = \mathbf{A}\,\mathbf{x}(n) + \mathbf{B}\,\mathbf{u}(n) + v$$

$$\mathbf{y}(n+1) = \mathbf{C}\,\mathbf{x}(n+1)$$

important special case: linear system. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are matrices.

- N-dimensional states $\mathbf{x} \in \mathbb{R}^d$
- discrete time steps $n \in \mathbb{Z}$ or $n \in \mathbb{N}$
- update operator is a function

  $$T: \mathbb{R}^d \to \mathbb{R}^d$$

Ordinary differential equations (ODEs)

$$\frac{d\mathbf{x}(t)}{dt} =: \dot{\mathbf{x}}(t) = T(\mathbf{x}(t))$$

Simplest case: autonomous system, i.e. no input, deterministic update

$$\dot{\mathbf{x}}(t) = T(\mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{y}(t) = R(\mathbf{x}(t))$$

Enriched with input $\mathbf{u}$: non-autonomous system. Further addition: output function R. Note: can't be made stochastic by simply adding noise term.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\,\mathbf{x}(t)$$

Important special case: linear system. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are matrices.

- N-dimensional states $\mathbf{x} \in \mathbb{R}^d$

- continuous time $t \in \mathbb{R}$ or $t \in \mathbb{R}^{\geq 0}$

- update operator is a function
  $$T: \mathbb{R}^d \to \mathbb{R}^d$$

- but now this operator specifies a rate of change, not the next state (as in all our models before)

Looks similar to iterated maps (deceptively). We will inspect such systems in more detail later

Stochastic differential equations
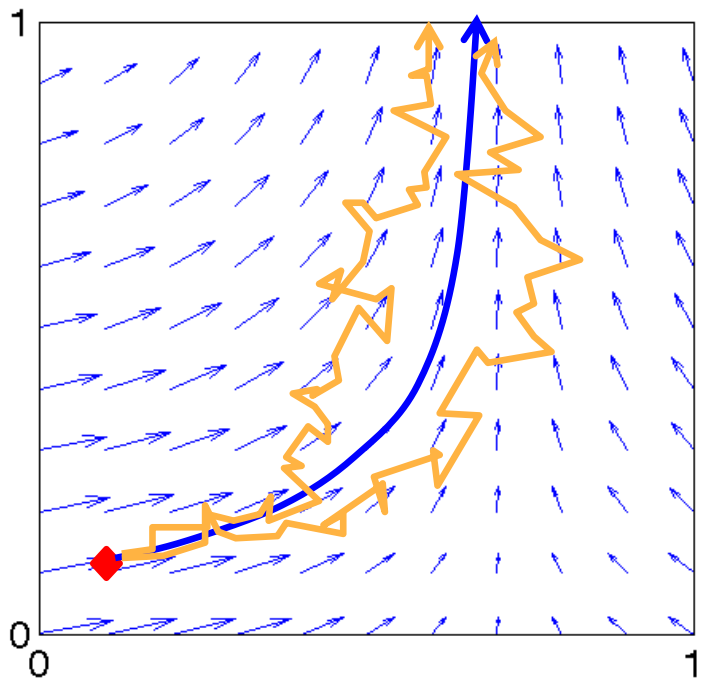
vector field guiding drift:
start point:
pure drift (deterministic) solution:
two realizations of stochastic process:

- Modeling goal: continuous-time processes of the kind "deterministic ODE-guided mechanism (drift component), all the time perturbed by Brownian motion kicks (diffusion component)

- Mathematically described by stochastic differentials of the kind

$$dX_t = \mu(X_t)dt + \sigma(X_t)dB_t$$

where drift is function μ, diffusion strength is σ and B is a Brownian motion trajectory

- Basic model type for the "hardcore theoretical physicist" modelers (in my perception: the "French school" of neural modeling)

Literature

Check out the Wikipedia page
https://en.wikipedia.org/wiki/Stochastic_differential_equation for entry
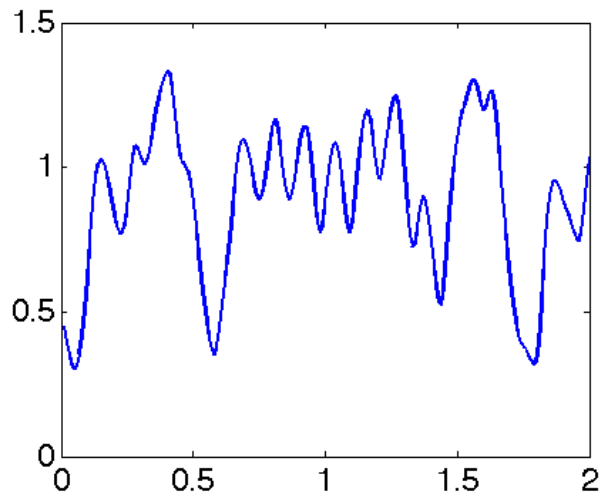points to concepts and literature.

A typical paper from the "French school":

M. Galtier, J. Touboul: Macroscopic Equations Governing Noisy Spiking
Neuronal Populations with Linear Synapses. PLOS One, November 13,
2013

Delay differential equations (DDEs)

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

Autonomous system case

A sample trajectory from the Mackey-Glass system

$$\dot{x}(t) = \frac{0.2\, x(t - 30)}{1 + x(t - 30)^{10}} - 0.1\, x(t)$$

- Derivative $\dot{\mathbf{x}}(t)$ depends on current state $\mathbf{x}(t)$ and state $\mathbf{x}(t\text{-}\delta)$ at previous time (or several previous times)

- Mathematically, the state at time t is not $\mathbf{x}(t)$ but it is the trajectory

$$\tau : [t - \delta, t] \to \mathbb{R}^N, \quad r \mapsto \mathbf{x}(r)$$

because all the $\mathbf{x}(r)$ in this interval co-determine the future.

- States are thus infinite-dimensional

- The induced dynamics can be extremely involved even for simple-looking 1-dim DDEs.

- The induced dynamics can be extremely involved even for simple-looking DDEs.

- DDE models arise naturally in neural dynamics modeling due to signal travel delays, e.g. along axons

Literature

A concise entry point to DDE literature and software:

Skip Thompson (2007) Delay-differential equations. Scholarpedia, 2(3):2367. www.scholarpedia.org/article/Delay-differential_equations
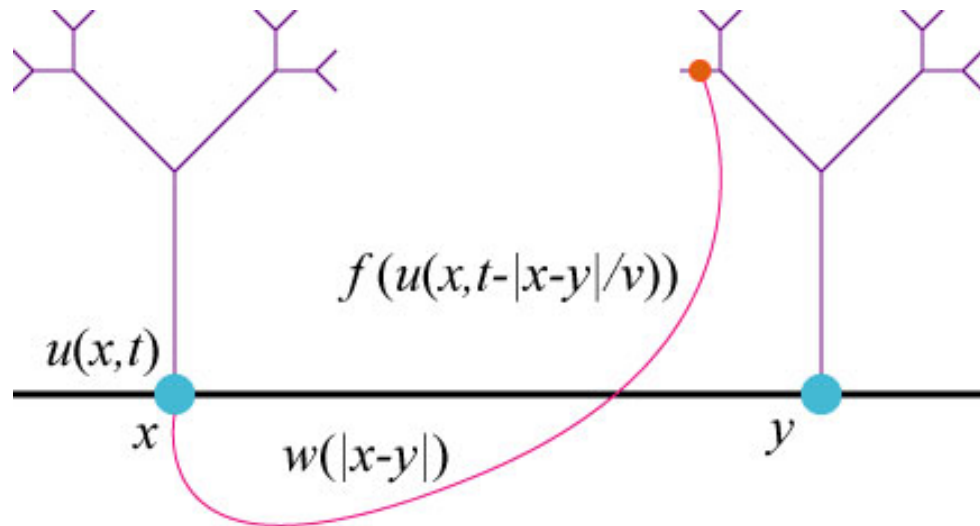
A tutorial on time delays in neural systems:

Campbell, S. A. (2007). Time delays in neural systems. In *Handbook of brain connectivity* (pp. 65-90). Springer Berlin Heidelberg. http://www.math.uwaterloo.ca/~sacampbe/preprints/brain.pdf

Partial differential equations (PDEs)

We skip them and immediately proceed to
the more general...

(Neural) field equations (FEs)

$f(u(x, t - |x-y|/v))$

$u(x,t)$

x

$w(|x-y|)$

y

$$c \frac{\partial u(x,t)}{\partial t} = -u + \int_{-\infty}^{+\infty} dy \; w(|x-y|) \, f(u(x-y, t - |y|/v))$$

- Motivation: model dynamics in spatially extended neural systems (e.g. travelling waves on cortical surfaces)
- Needs to take into account:
  - signal travel times
  - signal travel distances and field geometry
- Neural field equations hence combine DDEs, PDEs, integral equations

c time constant
u scalar field measurable (e.g. local firing rate)
x location in field (here: 1-dimensional "field")
w coupling strength profile (e.g. "Mexican hat")
v signal travel speed

http://www.scholarpedia.org/article/Neural_fields

Literature

A concise entry point to neural fields:

Stephen Coombes (2006) Neural fields. Scholarpedia, 1(6):1373
http://www.scholarpedia.org/article/Neural_fields

A tutorial text (I didn't read it):

Coombes, S., beim Graben, P., Potthast, R.: Tutorial on Neural Field Theory. In Coombes, S., beim Graben, P., Potthast, R., Wright, J. (eds.): Neural Fields. Springer Berlin Heidelberg, 2014

4. What is a State? ... and Takens' Theorem

The naive view

- a real (idealized, isolated) physical system evolves in continuous time

- continuous time is a line of "points in time"

- at each point in time t, the system is in a state $\mathbf{x}(t)$

- before the present point in time t, there is the past: $t' < t$; and thereafter, the future: $t'' > t$

- the system state $\mathbf{x}(t)$ at time t is physical reality: it **is** the system at time t, and it's **all** that there is

- a physical system has neither goals ($\mathbf{x}(t)$ doesn't "want" to change the future) nor memory ($\mathbf{x}(t)$ can't access the past)

I would call it the fundamental dogma of physics:

- states exist (they ARE THE reality)

- states endow the evolution of reality with the Markov property: future depends only on the present, not on the past

http://www.eoht.info/page/Billiard%20ball%20model

Some problems with that view

Problem 1: time itself

- theory of relativity: pointlike "presence" is not uniquely defined

- quantum mechanics: time itself might suffer from fine-granular uncertainty

I don't know enough physics to say more.

Problem 2: ontological status of state

- Kant: human thinkers don't have access to the thing-in-itself (Ding an sich)

- we can't know or understand the noumenonal (as opposed to phenomenal) essence of $\mathbf{x}(t)$

I don't know enough philosophy to dig deeper here.

Problem 3: freedom in modeling

- a neuron's state can be modeled e.g. as

  - a 1-dim binary "fires" vs. "fires not"

  - as 3-dim FitzHugh–Nagumo model state

  - ...

  - a 1000-dim compartment model

  - an infinite-dim spatiotemporal electrochemical pattern

- which model type / granularity is appropriate for what modeling goal?

- how are different models related?

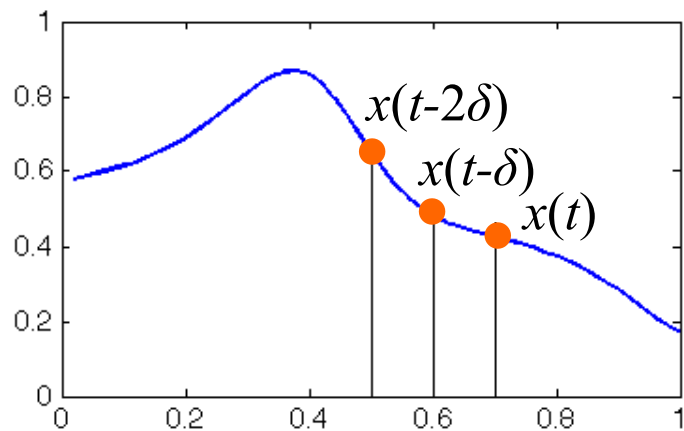That's food for thought for people like you and me.

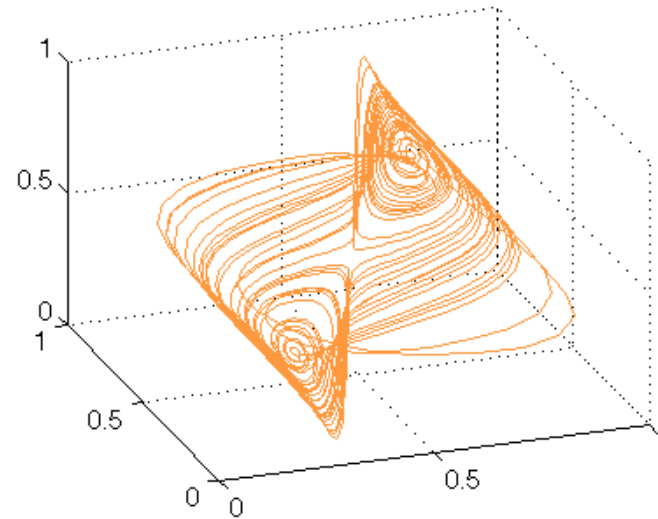# Creating state representations by delay embeddings

given: 1-dim timeseries $x(t)$
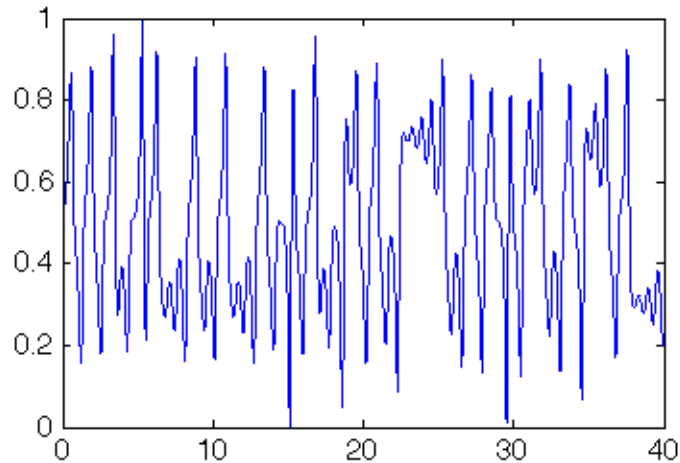
turn into 3-dim timeseries
$$\mathbf{x}(t) = (x(t), x(t\text{-}\delta), x(t\text{-}2\delta))'$$

plot the 3-dim timeseries $\mathbf{x}(t)$

- we can artificially transforme a 1-dim state into an m-dim state by time-delay embedding

- no information added

- what is the "real" system state?

Takens' theorem (simplified)

Preparation: A *d*-dimensional manifold is a subset of R^n that locally is *d*-dimensional. Example:

- unit sphere surface: 2-dim manifold in R^3

Theorem of Takens

Given: A smooth vector field F on a *d*-dimensional manifold M, a map $\varphi : M \to R$.

- The vector field F induces a dynamics $\mathbf{m}(t)$ on M, and $\varphi(\mathbf{m}(t)) =: \mathbf{o}(t)$ gives 1-dim observation timeseries.

- Consider the $2d+1$-dimensional time-delay embedding state dynamics

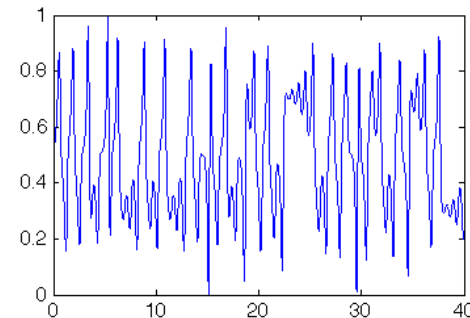$$\mathbf{e}(t) = (\mathbf{o}(t), \mathbf{o}(t\text{-}\delta), ..., \mathbf{o}(t\text{-}2\delta)).$$

Then, $\mathbf{e}(t)$ evolves on a *d*-dimensional manifold $M' \subseteq R^{2d+1}$, and the vector field F' associated with this evolution is homeomorphic to F.
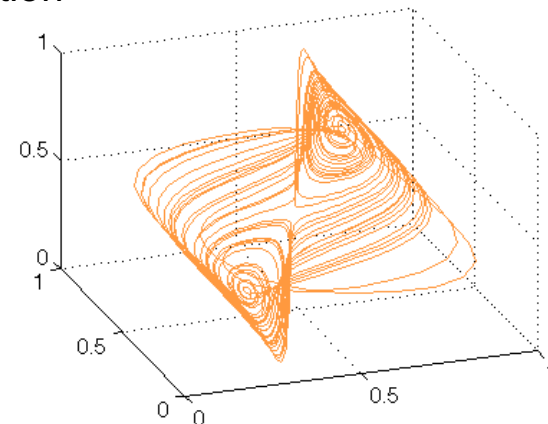

$\mathbf{m}(t)$
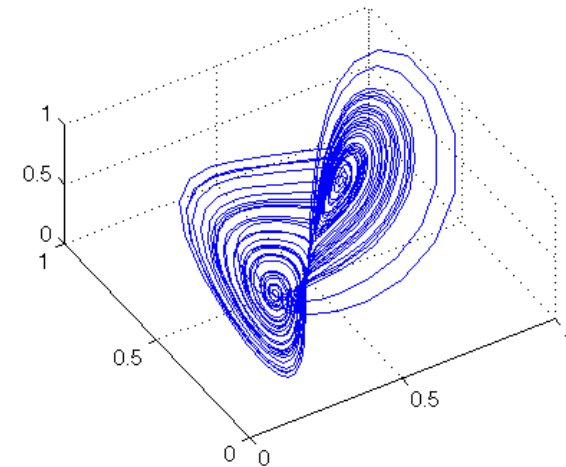
1-dim
observation

$\mathbf{o}(t)$

delay-embedded
reconstruction

$\mathbf{e}(t)$

Takens' theorem, explanations

- "Homeomorphic" means: the same up to smooth "space distortions" ("rubber-sheet transformations")

- Important characteristics of the reconstruction $\mathbf{e}(t)$ are identical to $\mathbf{m}(t)$.

- Specifically, Lyapunov exponents (which quantify stability or chaoticity) are preserved.

- The grand message: 1-dim observations of ODE systems, traced over time, reveal everything that is essential about the invisible, generating high-dim system.

Teaser question: what is the real / true state of an ODE-governed system? or the real / true state dimension?

$\mathbf{m}(t)$

$\mathbf{o}(t)$

$\mathbf{e}(t)$
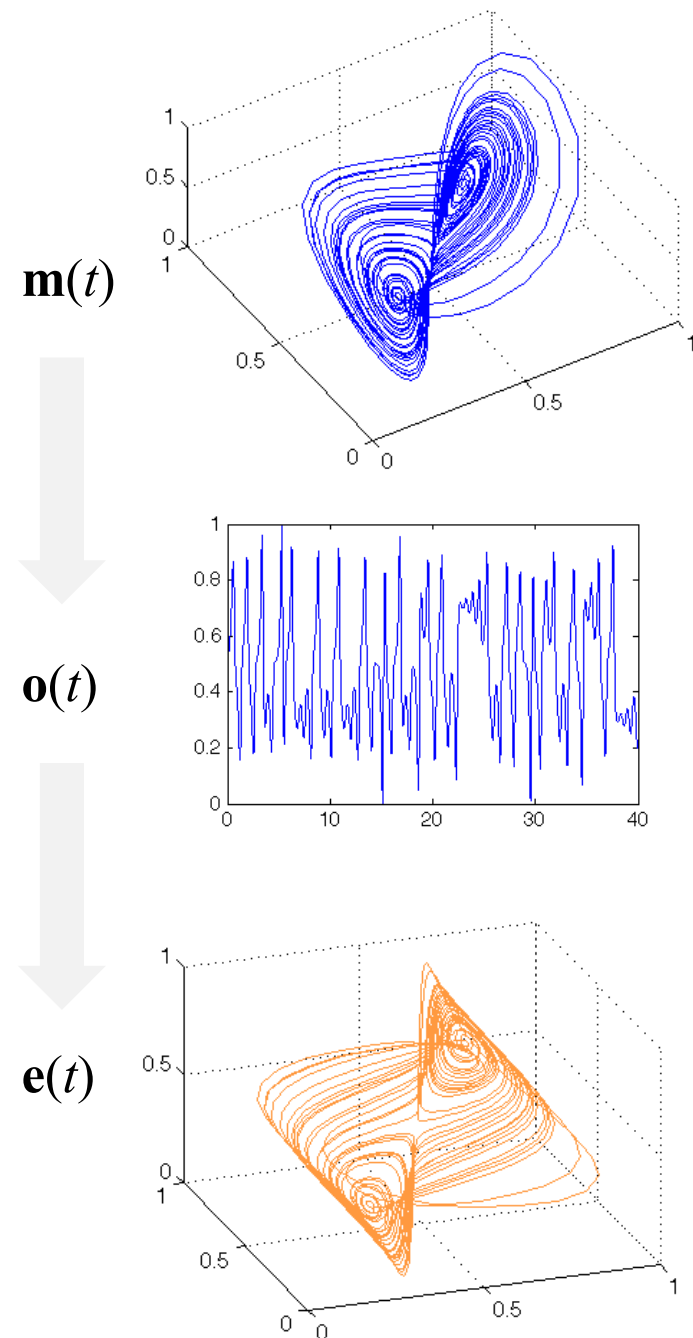
Takens' theorem, comments

- TT gives an analytical justification for reconstructing complex systems from low-dim observations

- Specifically, TT can be used to detect and quantify chaos on the basis of data from a few scalar observation channels

- Use in the cognitive neurosciences: quantify degrees of chaos / complexity measures of underlying high-dimensional neural activity from observed timeseries measurements

- CAVE: when there is noise involved, analyses must be carried out with the greatest mathematical care. It is difficult to distinguish noise from high-dimensional chaos.
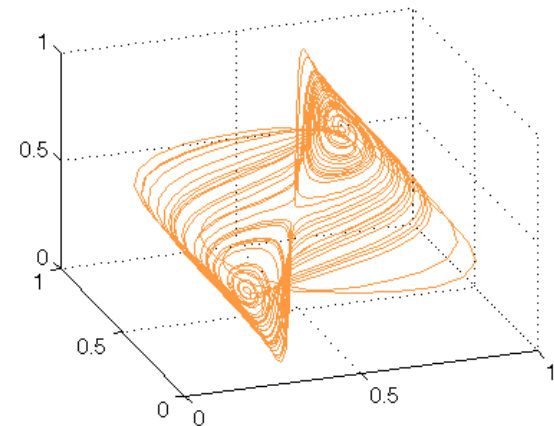
$\mathbf{m}(t)$
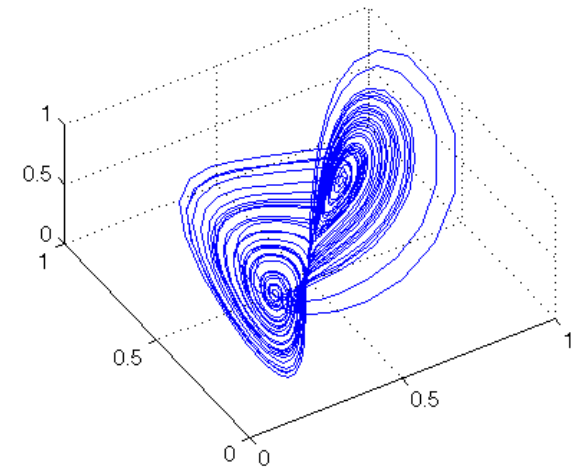
$\mathbf{o}(t)$
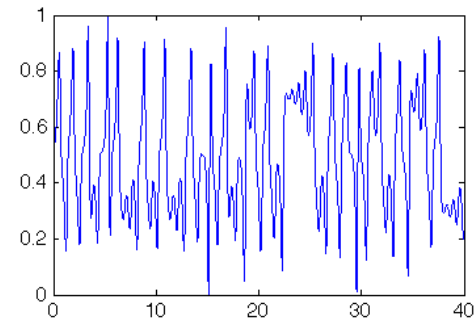
$\mathbf{e}(t)$

Literature

The original paper of Takens (>15000 Google cites):

Florin Takens, Detecting strange attractors in turbulence. In Rand, D.A. and Young, L.-S. (eds.), Dynamical Systems and Turbulence. Lecture Notes in Mathematics 898, Springer Verlag 1991, 366-381

A generalization to far larger classes of systems:

Stark, J., Broomhead, D. S., Davies, M. E., Huke, J., Takens embedding theorems for forced and stochastic systems. Nonlinear Analysis, Theory, Methods & Applications 30(8), 1997, 5303-5314

A super-carefully done video of "Takenizing" the Lorenz attractor is at
https://www.youtube.com/watch?v=6i57udsPKms

Back to discussing states...

state $\mathbf{x}(0)$

- Takens theorem reveals a view how "states" are the same as "collect some observation info over previous history"

- Other models likewise have "time-accumulated info-states" e.g.

  - higher-order Markov chains

  - transversal filter systems (introduced later today)

  - delay differential equations

  - general formalism of stochastic processes where transition law generates next-step distribution on all previous observations, as in

$$P(X_{n+1} = a \mid X_0 = b_0,...,X_n = b_n)$$

where $(b_0, ..., b_n)$ can be regarded as state.

-4 -3 -2 -1 0 1 2 3 4

n
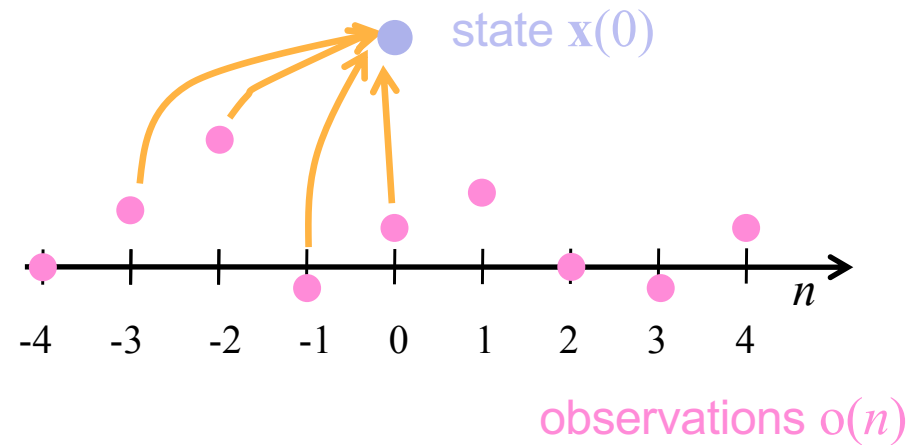
observations $\mathrm{o}(n)$

Turning the tide of time

- In Takens etc. a state could be interpreted / conceived / reconstructed from / as previous observation history sniplets. One could call this "history memory states"

state $\mathbf{x}(0)$

n

-4 -3 -2 -1 0 1 2 3 4

observations o(n)
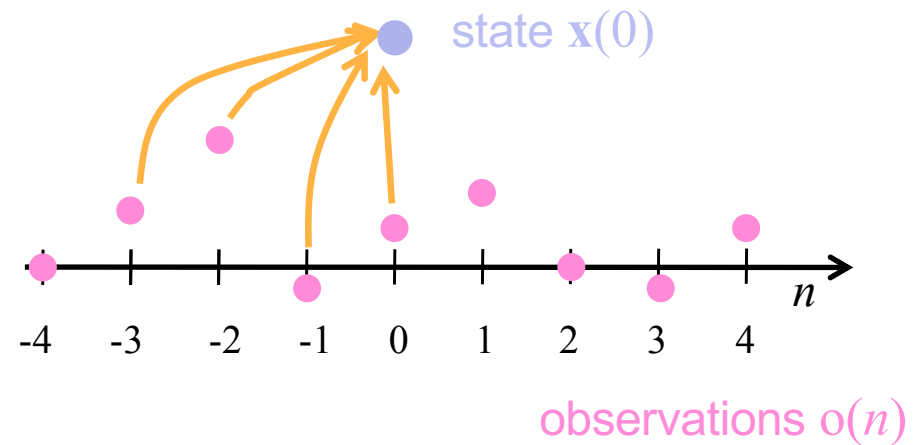
- One can also construe states as "information states" which consist of the information necessary to predict the future. One could call this future-predictive states.

state $\mathbf{x}(0)$

n

-4 -3 -2 -1 0 1 2 3 4

- This is possible and has been done in machine learning and automata theory:
    - observable operator models
    - predictive state representations
    - multiplicity automata
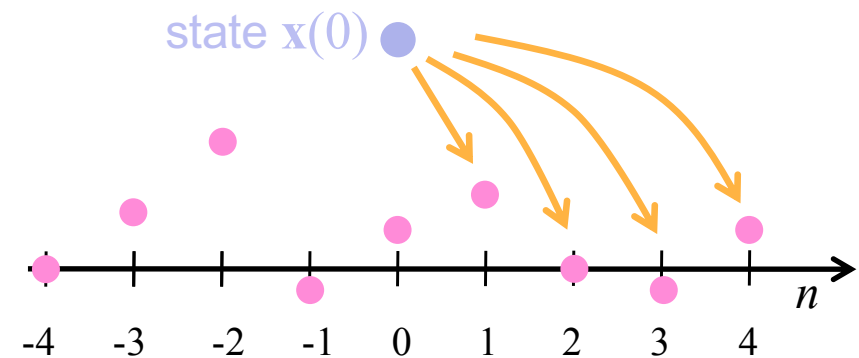
" ... a state of a system at any given time is the information needed to determine the behaviour of the system from that time on." L. A. Zadeh, 1969

Literature

The Zadeh quote is from an article that gives a general, abstract, formal definition of states-as-future-encodings (in a context of classical systems engineering):

L.A. Zadeh. The concept of system, aggregate, and state in system theory. In L.A. Zadeh and E. Polak, editors, System Theory vol 8, 3 - 42. McGraw-Hill, New York, 1969.

A survey and unification of observable operator models, predictive state representations, and multiplicity automata is in

Thon, M., Jaeger, H. (2015): Links Between Multiplicity Automata, Observable Operator Models and Predictive State Representations - a Unified Learning Framework. Journal of Machine Learning Research 16(Jan):103−147 http://jmlr.org/papers/volume16/thon15a/thon15a.pdf

5. State-Free Modeling of Temporal Systems

The classical engineering view on "systems"

$u(n\text{-}1)$ $u(n)$ $u(n\text{+}1)$

· · ·   · · ·

$y(n\text{-}1)$ $y(n)$ $y(n\text{+}1)$

" system", "filter"

H
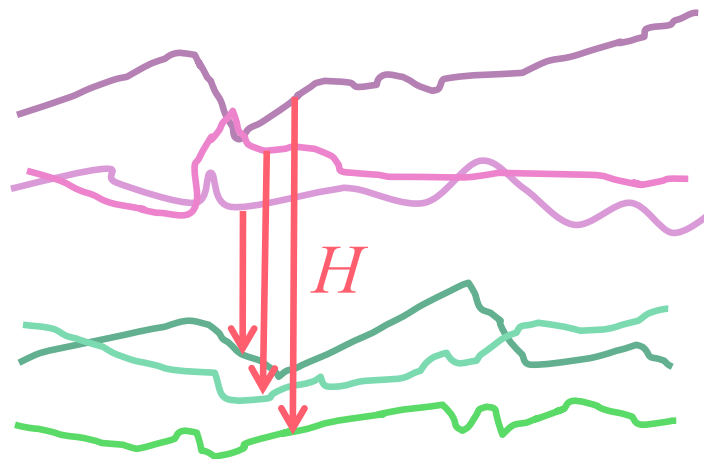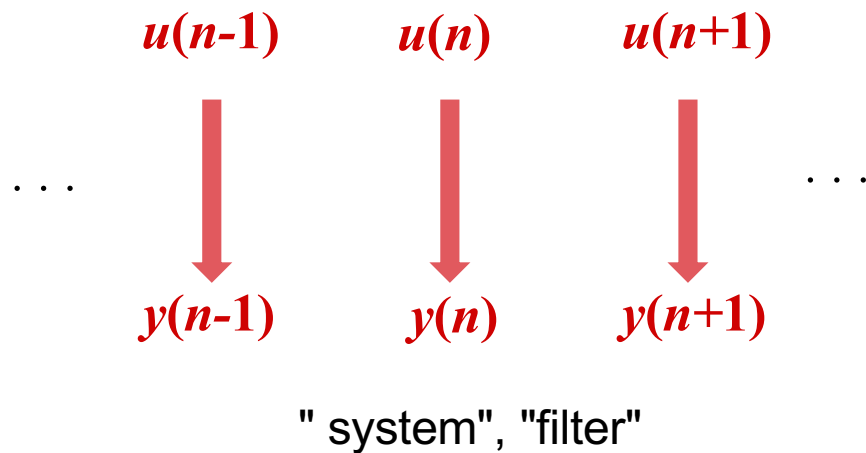
- the classical perspective taken by signal processing and control engineers

- focus on the mathematical relationships between input signals $u(n)$ and output signals $y(n)$

- A signal s (input u or output y) is a mapping from time to (typically scalar) values:

  $$s: \mathbb{Z} \longrightarrow \mathbb{R} \text{ (or } \mathbb{C}) \quad \textit{discrete time} \text{ or}$$

  $$s: \mathbb{R} \longrightarrow \mathbb{R} \text{ (or } \mathbb{C}) \quad \textit{continuous time}$$

  (suitably constrained, e.g. finite variance)

- Signals are just "time series"

- Let S be the set of signals (discrete or continuous time, suitably constrained)

- A filter (or system) is a functional

  $$H: \text{S} \longrightarrow \text{S}$$

- A filter is just a value map from input timeseries to output timeseries ... *any* such map, with no "mechanism" implied

Continuous vs. discrete-time signals and systems
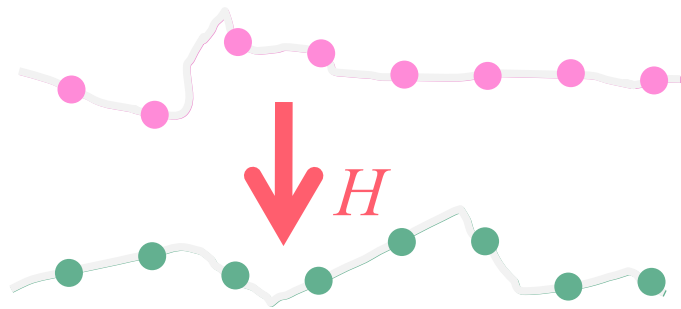
- The historically earlier theory concerned continuous-time analog signals and filters.

- With the advent of modern computers, discrete-time digital signals and filters became equally (if not more) important.

- Analog signals turn into digital signals by sampling (not to be confused with "sampling" in the sense of statistics).

- An important aspect in signal engineering: what information is preserved/lost in sampling.

- Nyquist-Shannon (et al.) sampling theorem: lossless sampling if sampling rate is at least $2B$ samples/second, where B is highest-frequency component in analog signal

- Mathematical analyses of continuous vs. discrete-time signals and systems are related but not 1-1 translatable into each other.

I will only consider discrete-time signals $s(n)$ in what follows.

Basic concepts and classes of signals and systems

$$H(\alpha\, u_1 + \beta\, u_2) = \alpha\, H(u_1) + \beta\, H(u_2)$$

- **Linear** systems transform linear combinations of input signals into linear combinations of their respective outputs.

$$\big(H(u)\big)(m) = y(m) \Rightarrow$$

$$\big(H(u(n-D))\big)(m) = y(m-D)$$

- **Time-invariant** systems transform time-shifted inputs into time-shifted outputs.

- **"LTI"**: linear, time-invariant systems. The bulk of systems considered by signal engineers.

$$y(n) =$$

$$f(u(n), u(n-1), ..., y(n-1), y(n-2), ...)$$

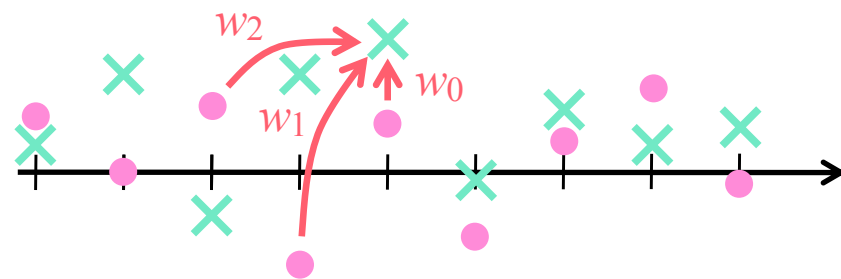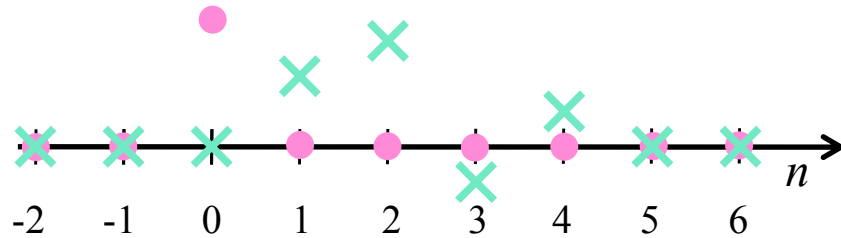- A system is **causal** if current output depends only on previous history

$$\forall n : \big|u(n)\big| \le A \Rightarrow$$

$$\exists B : \forall n : \big|H(u)(n)\big| \le B$$

- A system is **bounded-input bounded-output stable** (BIBO stable) if bounded inputs are transformed to bounded outputs

LTI systems

$$\delta(n) = \begin{cases} 1, & \text{if } n = 0 \\ 0, & \text{else} \end{cases}$$

- The unit impulse δ is the signal that is zero everywhere except at $n = 0$, where it is 1

- An LTI system H is uniquely determined by its impulse response $H(\delta)$.

- Finite-impulse-response (FIR) filters: $H(\delta)$ returns to zero after a finite number of steps. Infinite-impulse-response (IIR) filters: never-ending "excitation".

- FIR filters can be understood as a moving average:

$$y(n) = w_0\, u(n) + w_1\, u(n-1) + ... + w_k u(n-k)$$

- If auto-regression (output feedback) is included, IIR filters result (ARMA filters = auto-regressive moving average)

$$y(n) = w_0\, u(n) + w_1\, u(n-1) + ... + w_k u(n-k) +$$
$$v_1\, y(n-1) + ... + v_l y(n-l)$$

- FIR filters are BIBO-stable; ARMA filters can be unstable, e.g., $y(n) = 2\, y(n-1)$

Linear systems: frequency-domain analysis

Fourier transform of continuous-time signal $s(t)$:

$$F(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} s(t)\, dt, \ \omega \in \mathbb{R}$$

Laplace transform of $s(t)$:

$$F(z) = \int_{-\infty}^{\infty} e^{-z t} s(t)\, dt, \ z \in \mathbb{C}$$

Fourier transform of discrete-time signal $s(n)$:

$$F(\omega) = \sum_{n=-\infty}^{\infty} e^{-i\omega n} s(n), \ \omega \in [0, 2\pi]$$

z-transform of $s(n)$:

$$F(z) = \sum_{n=-\infty}^{\infty} z^{-n} s(n), \ z \in \mathbb{C}$$

- Main mathematical tool for working with linear systems: "frequency domain" analysis. Signals are transformed from their original time domain to the frequency domain by Fourier transforms or generalizations thereof (Laplace transform, z-transform)

- One of the reasons why frequency domain transforms are so eminently useful:
  - Convolution (applying a filter to a signal) in the time domain becomes multiplication in the frequency domain
  - Deconvolution (inverting a filter, basic operation in controller design) becomes division

- These transforms only work for linear systems

Signals and systems, engineering tradition: comments

- A venerable, century-old, comprehensive, powerful, elegant theory – *in the linear case*. 90% – 100% of the material in a random engineering textbook will be about linear systems.

- Many connections to neuroscience:

  - Supplier of intuitions and functional models of neural information processing, especially in

    - pattern recognition (auditory and visual)

    - motor control

  - Huge "toolbox" for neural signal analysis

  - Some shared historical roots (Norbert Wiener / Cybernetics / feedback control)

  - Background theory context for neural networks in nonlinear control (K. S. Narendra, Michael Jordan, Eduardo Sontag)

Literature

Signal processing and control theory is one of the biggest fields of engineering and science in general, and there is a plethora of textbooks and online tutorial materials. Here is an adhoc sample:

Undergraduate-level lecture notes, elementary, detailed: Stanley Chan, Class notes for signals and systems, UCSD
scholar.harvard.edu/stanleychan/files/**note_**0.pdf

Nonlinear signal processing and control is less "classical" and more of a specialization area for engineers.
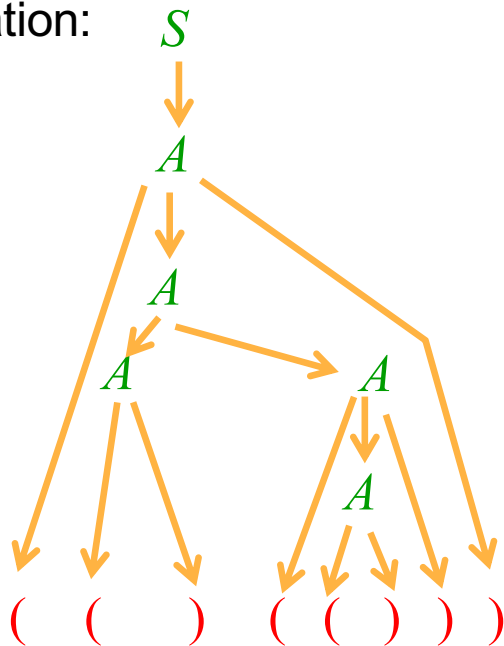
Context-free grammars

Example

$V = \{S, A\}$

$T = \{(,)\}$

$P: S \rightarrow A, A \rightarrow (A), A \rightarrow (\ ),$

$\quad A \rightarrow AA$

A derivation:

Due to this derivation, the word $(()(()))$
is in this grammar's language.

A context-free grammar (CFG) is defined by

- a finite set V of variables,

- a start variable $S \in V$,

- a finite set T of terminals

- a finite set of production rules
   $$P \subseteq V \times (V \cup T)^*$$

- A CFG defines a language, i.e. a set of finite symbol sequences (words) over T through derivations

Grammars, automata, Chomsky hierarchy

The Chomsky hierarchy

- simplest: regular languages, linear grammars, DFAs (no memory)

$$\text{IN}$$

- next simple class: context-free languages, CFGs, pushdown automata (single-stack memory)

$$\text{IN}$$

. . .

$$\text{IN}$$

- most complex class: recursive languages, unrestricted grammars, Turing machines (infinite tape memory)

- Grammars can be made more expressive by allowing more complex production rules

- Always a grammar defines a language by derivable words

- Grammars define internal structure of symbol sequences

- Pioneer since the 1950's: linguist Noam Chomsky (for some years the most cited scientific author, period)

- each Chomskian grammar type comes with an equivalent automaton model

- more expressive grammar: more powerful memory added to DFA

- Today a (maybe *the*) main tool of the theory of programming languages

Exponential information washout: the default behavior in timeseries

$$R(d) = \mathrm{cov}(x(t),\, x(t+d))$$
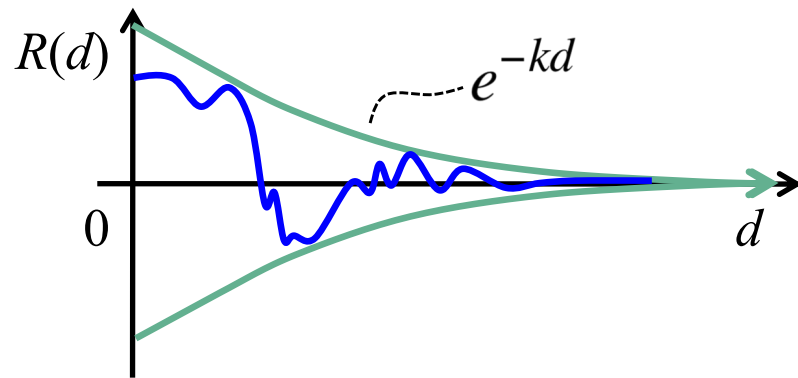
- Consider the autocovariance $R(d)$ of a numerical timeseries $x(t)$ generated by a stationary process

- Often $R(d)$ has an exponentially decreasing envelope: $|R(d)| \leq \exp(-kd)$

- Intuitive explanation: in system update laws like
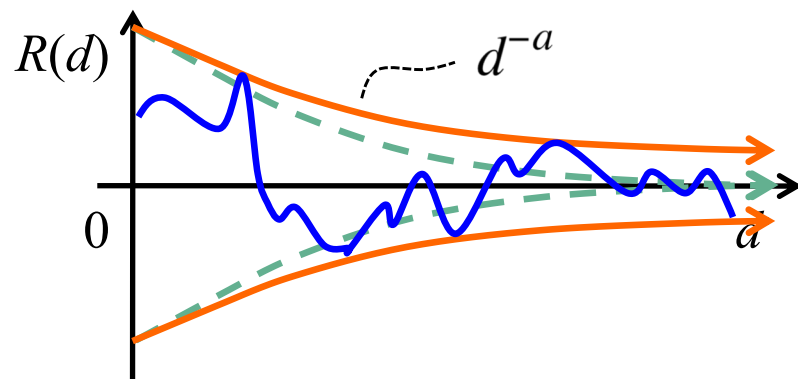$$\dot{x}(t) = T(x(t), u(t)) \quad \text{or}$$
$$x(n+1) = T(x(n), u(n+1))$$

  the continuing inflow of input signal energy $u(t)$ replaces signal energy $x(t)$ at a certain average rate

- Many other range-dependent characteristics show similar exponential decay with interaction distance

"Long-range" interaction: power laws

$$R(d) = \text{cov}(x(t), x(t+d))$$

$R(d)$

d^{-a}

0

a

- Sometimes however one witnesses a slower, polynomial decay of interaction strength with interaction distance:

$$R(d) \leq d^{-a} \text{ for some } a > 0$$

- Terminological footprint:
  - `power law`
  - `long-range interaction`
  - `fat tail`
  - $1/f$ `law (origin: frequency spectra in Fourier transforms)`

- Intuition: certain signal components in the past are selected (structure formation effects) and preserved over time (memory effects)

Power-law behavior and grammars
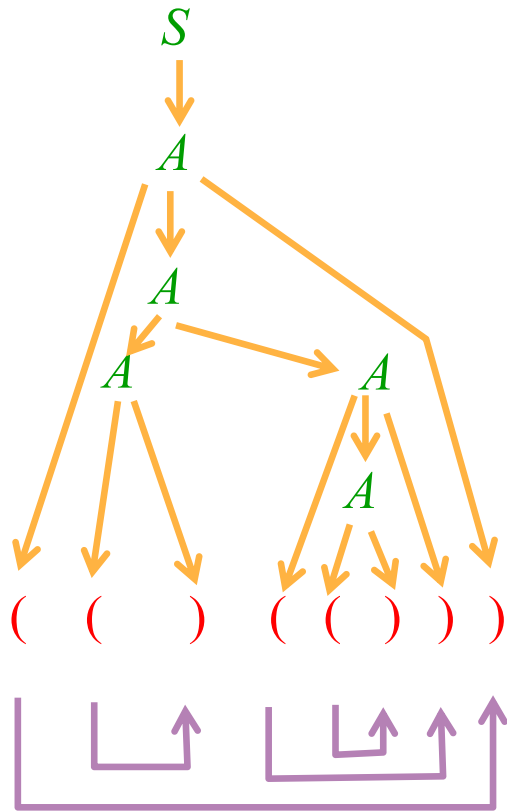
Nested dependencies, long-range effects at different timescales, no single characteristic timescale

- Grammars are a prime tool to describe long-range dependencies in symbol sequences

- Continuous-time and/or continuous-value timeseries can be made symbolic by discretization

- Timeseries with power-law properties often exhibit structural characteristics like

  - multiple relevant timescales

  - hierarchical organisation

  - self-similarity, fractal properties

  - (close to) chaotic dynamics

- Rich phenomenology, rich literature

- Human-generated "intelligent" output typically has power-law properties: speech, motion patterns

Literature

Grammar formalisms are introduced at length in any standard textbook for theoretical computer science, e.g.

Hopcroft, John E. , Motwani, Rajeev, and Ullman, Jeffrey: *Introduction to Automata Theory*, 2nd edition. Addison-Wesley 2000

A much-cited paper on long-range (grammatical) structure in nonlinear dynamics at the onset of chaos:

Crutchfield, J.P. and Young, K.: Computation at the Onset of Chaos. In: Zurek, W.H. (ed.), Complexity, Entropy, and the Physics of Information, Addison-Wesley 1990, 223-269
http://csc.ucdavis.edu/~cmg/papers/CompOnset.pdf

Advertising *pro domo*: An outstanding BSc thesis that explores the connections between grammatical structure and "left-to-right" temporal generation of symbol sequences:

Djolonga, J.: Comparison of the Expressive Powers of Weighted Grammars and OOMs. BSc thesis, Jacobs University Bremen 2011.
https://www.ai.rug.nl/minds/uploads/Thesis_jdjolonga.pdf

This quite non-technical book (in German) is all about structure in timeseries:

W. Ebeling, J. Freund, F. Schweitzer: Komplexe Strukturen: Entropie und Information. Teubner Stuttgart Leipzig 1998

6. Qualitative Theory of DS: Attractors, Bifurcations, Chaos

Context

- Global intuition: if a DS exhibits a "stable dynamical pattern", it is likely in an attractor. Transitions from one pattern to another: can often be modelled as bifurcation.

- Attractors and bifurcations (A & B) were increasingly studied in (pure) maths and theoretical physics since about the 1960-ies

- Field was boosted by the computer-simulation enhanced study of fractals and chaos, thoroughly permeating many scientific (and popular) fields since about the late 1970-ies

- Terminological footprint: `nonlinear dynamics`, `dynamical systems theory`, `self-organization`, `pattern formation`, `emergent behavior`, `Cybernetics`, `Synergetics`, ...
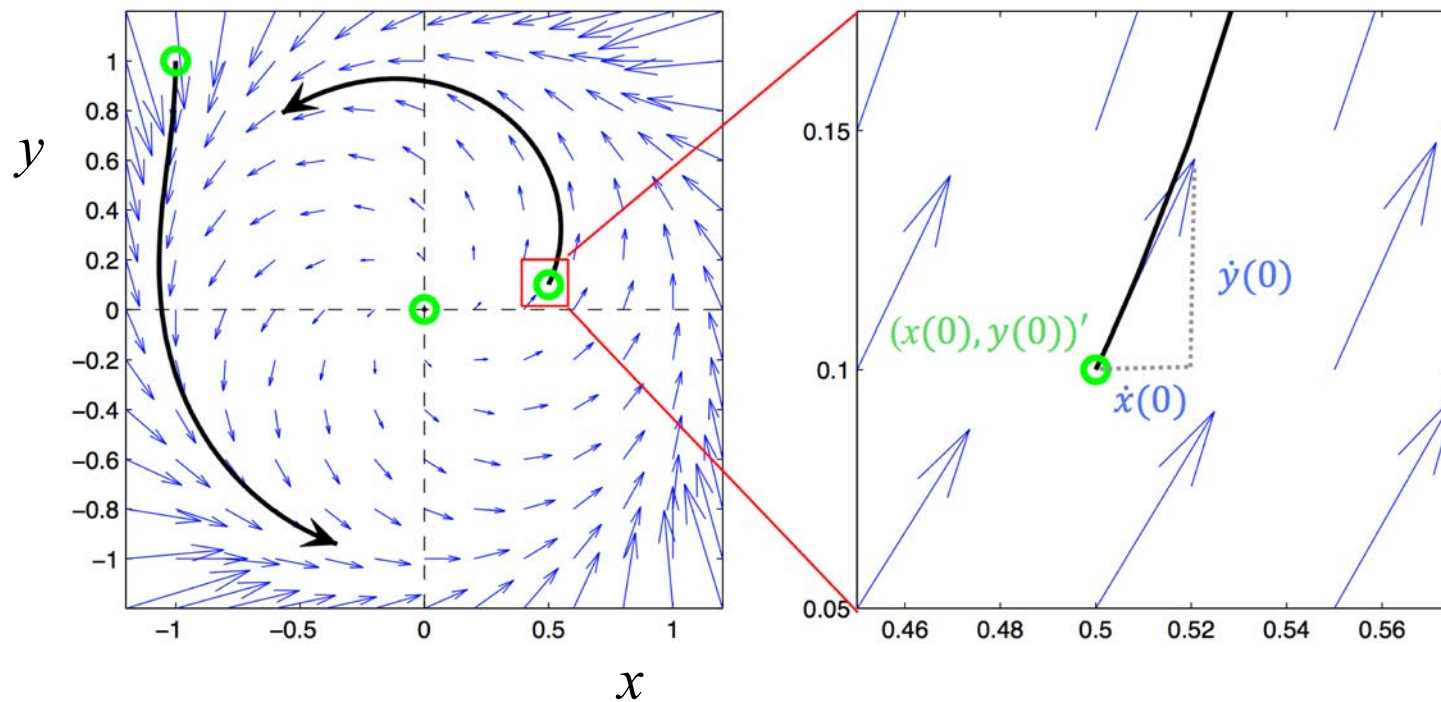
Personal opinion: though clearly important and insightful, this set of tools and metaphors is (still) too weak and narrow to serve as the only reference maths for modeling neural dynamics.

Our 2-dimensional tutorial example

System equation in x, y coordinates

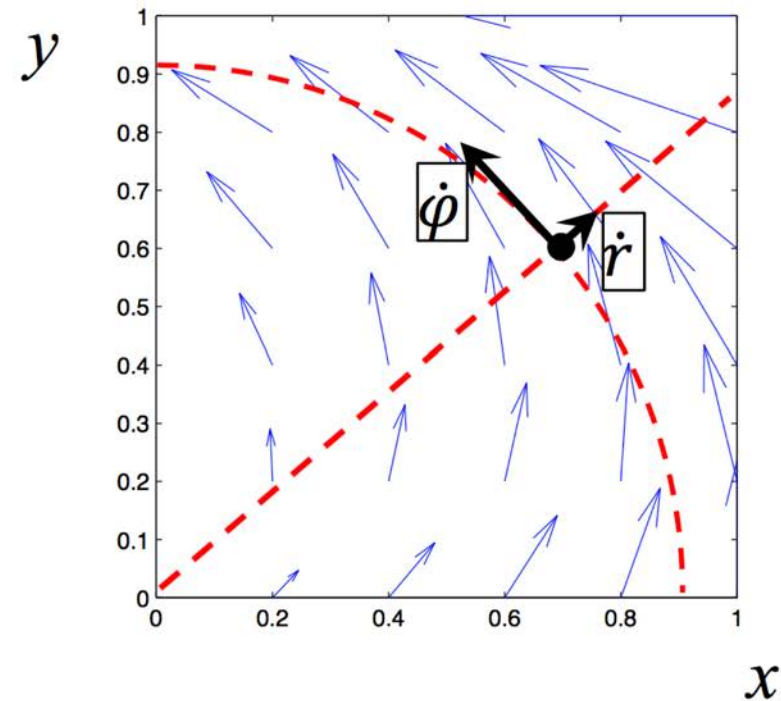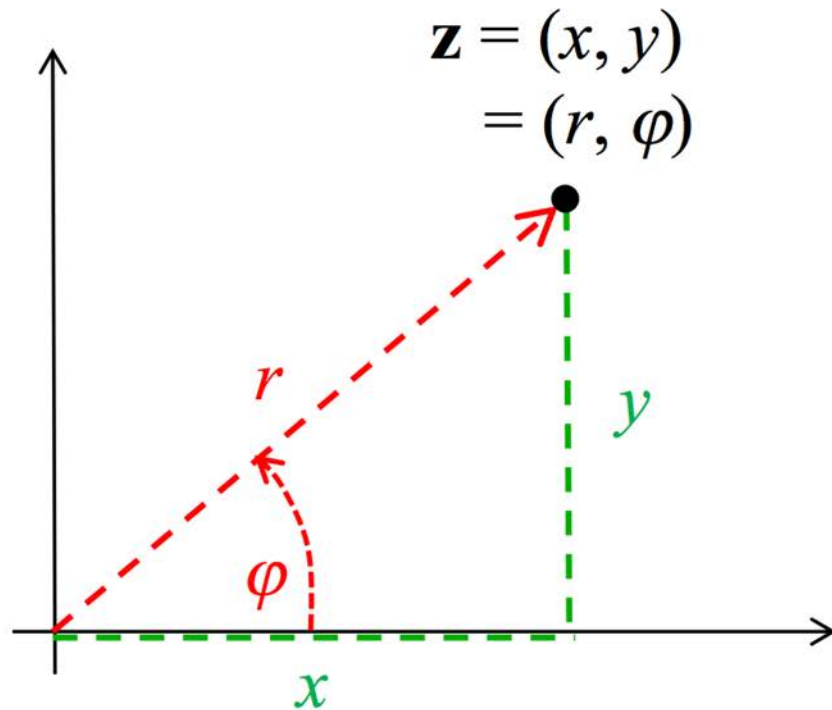$$\dot{x} = -x(x^2 + y^2)^3 + x - y$$
$$\dot{y} = -y(x^2 + y^2)^3 + y + x$$

- At each point (x, y) gives a vector $(\dot{x}, \dot{y})$
- All these vectors together give a vector field
- State motion follows the direction of vectors with speed given by length of vector

Same system in polar coordinates

$$r = \sqrt{x^2 + y^2}, \theta = atan2(x, y)$$

System equations now look much easier:

$$\dot{r} = -r^3 + r$$
$$\dot{\varphi} = 1.$$

Literature / Resources

In the neurosciences, an influential proponent of DS methods is Bard Ermentrout, http://www.pitt.edu/~phase/ . He wrote an ODE solver-simulator with an emphasis on graphical output, called XPP or XPPAUT, which is available for all operating systems (https://sites.pitt.edu/~phase/bard/bardware/xpp/xpp.html). For DS-oriented explorations this tool is more convenient than e.g. Matlab because (i) it is more interactive, good for hands-on exploration of parameter settings, and (ii) its graphical output standardly displays elements typical for DS theory analyses, which in Matlab you would have to explicitly program. The graphical user interface of XXPAUT needs some time to get used to however.

A recent book by Ermentrout & Terman:

B. Ermentrout, D. H. Terman: Mathematical Foundations of Neuroscience. Springer (Interdisciplinary Applied Mathematics series), 2010
Despite its title, it's not really covering math foundations in general but focusses on DS.

 A very well-written tutorial paper explaining a number of dynamical phenomena which are particularly relevant (and popular) in neuroscience models is

Miller P. (2016). Dynamical systems, attractors, and neural circuits. *F1000Research*, 5, F1000 Faculty Rev-992. doi:10.12688/f1000research.7698.1
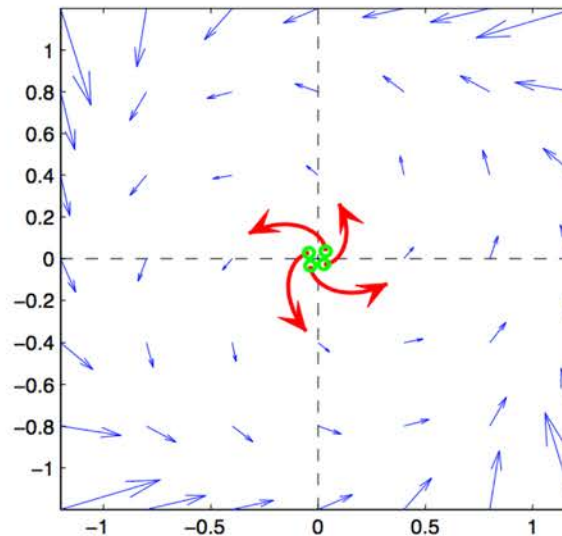
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4930057/

Trajectories and phase portraits

$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$

A

B
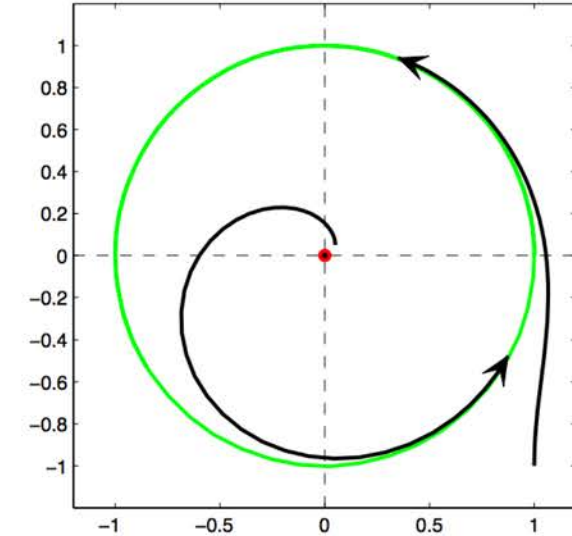

C
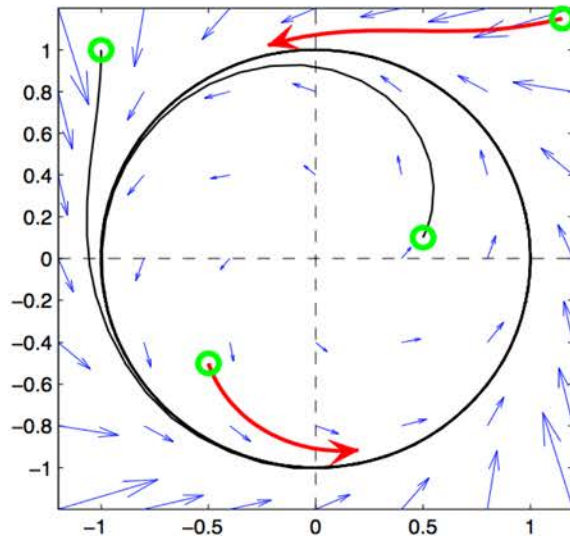
- Following the vector field from any starting point o gives a trajectory.
- Starting outside the unit circle lets state history „spiral in" toward unit circle.

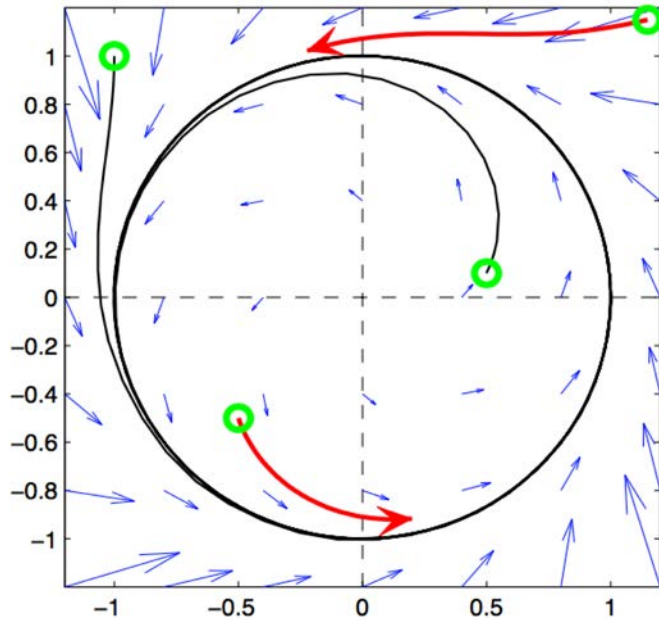Starting close to the origin leads to „spiraling outward" toward unit circle.

- Plotting several characteristic trajectories gives a phase portrait.
- Starting exactly a the origin leads to standstill: no motion.

Analysis 1

$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$

Observation: (almost) all trajectories home in on a limit cycle. -- Why?
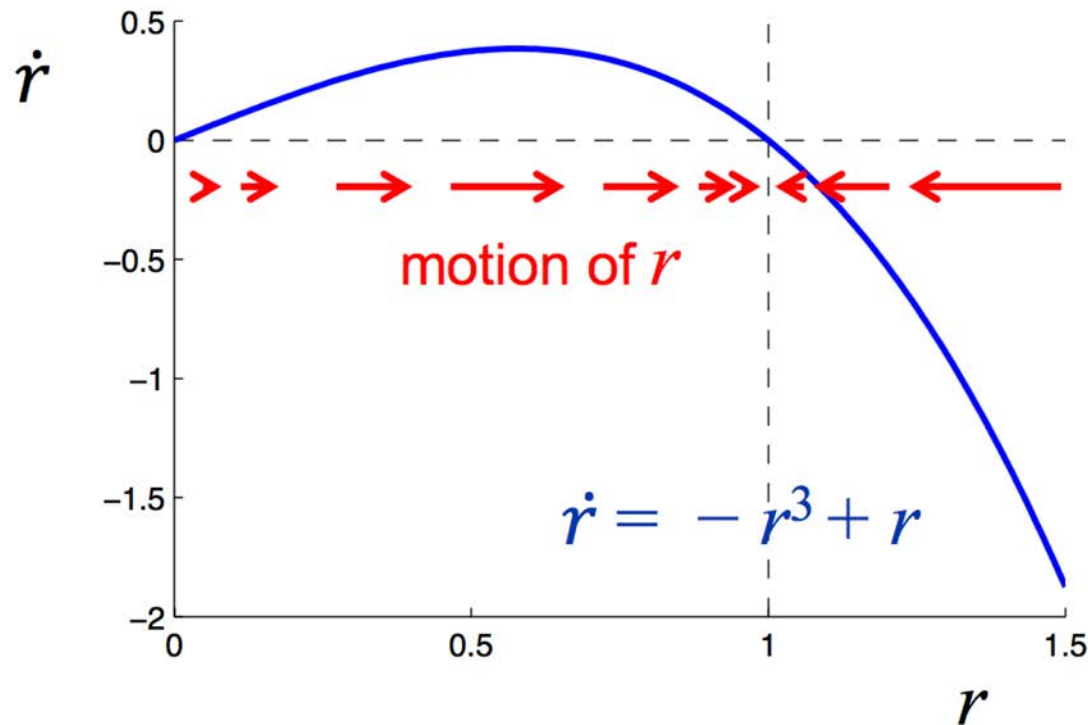
- This example is easy to analyse because the equations for r and θ are decoupled (r does not appear in equation for θ and vice versa).

- $\dot{\theta} = 1$ implies that all motion revolves around the origin with constant angular velocity.

- The key lies in the radial component of the dynamics, $\dot{r} = -r^3 + r$.
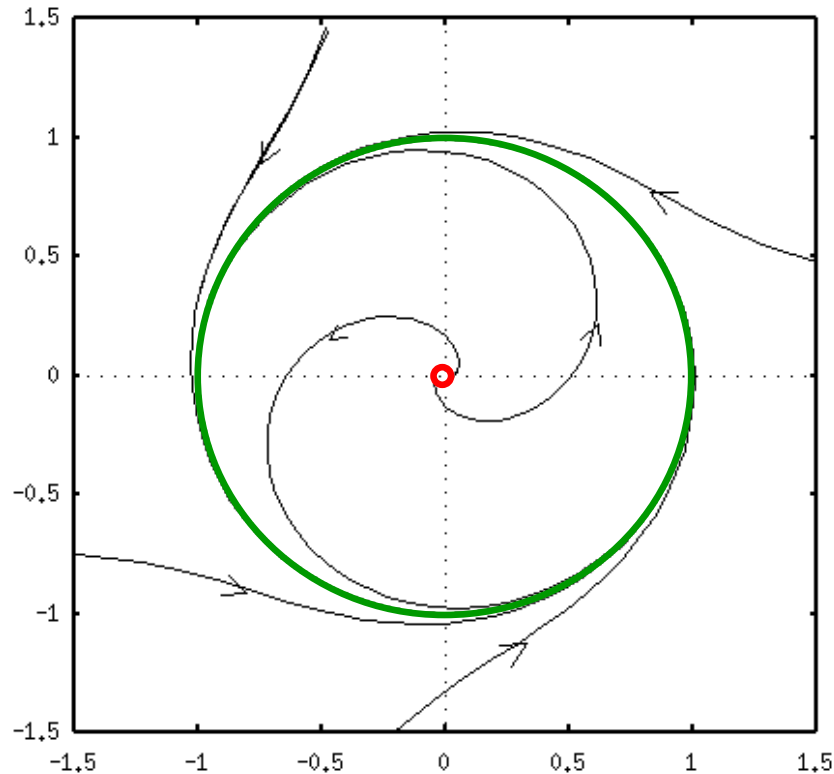
Analysis 2

Analysis of $\dot{r} = -r^3 + r$
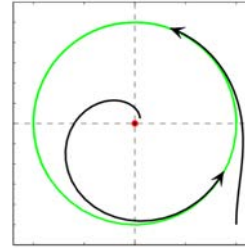
$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$

motion of r

$\dot{r} = -r^3 + r$

- For $0 < r < 1$, we have $\dot{r} > 0$, so r will grow with time.

- For $1 < r$, we have $\dot{r} < 0$, so r will shrink with time.

- For $r = 0$ and $r = 1$, we have $\dot{r} = 0$, so r will stay fixed at these values.

- $\longrightarrow$ indicate motion of r.

Two special trajectories

$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$

Start from $(0, 0)$: Trajectory will forever remain in $(0, 0)$. This is a fixed point of the dynamics.

This fixed point is unstable: even the smallest perturbation will drive trajectory away from it. The origin is a repellor.
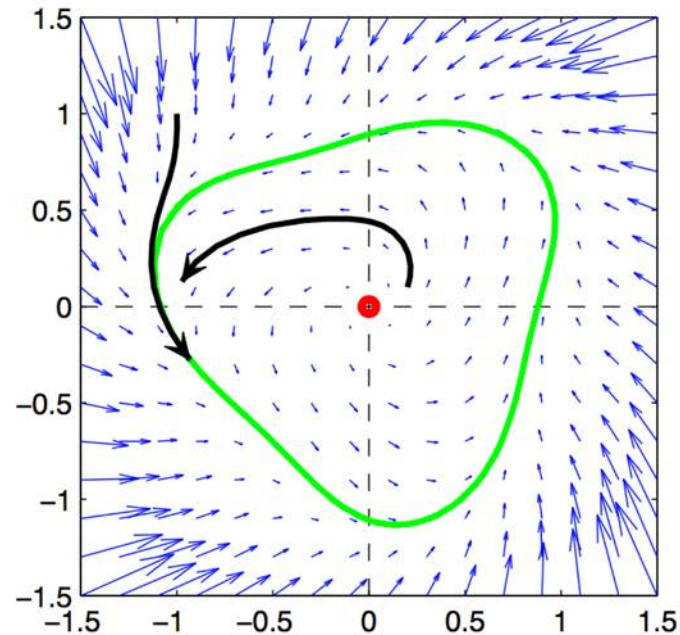
Start from anywhere on the unit circle: the trajectory will forever revolve around the unit circle.

This periodic orbit is stable: when perturbed, the trajectory will asymptotically return to the cycle. It is a limit cycle, a cyclic attractor.

Surprise!
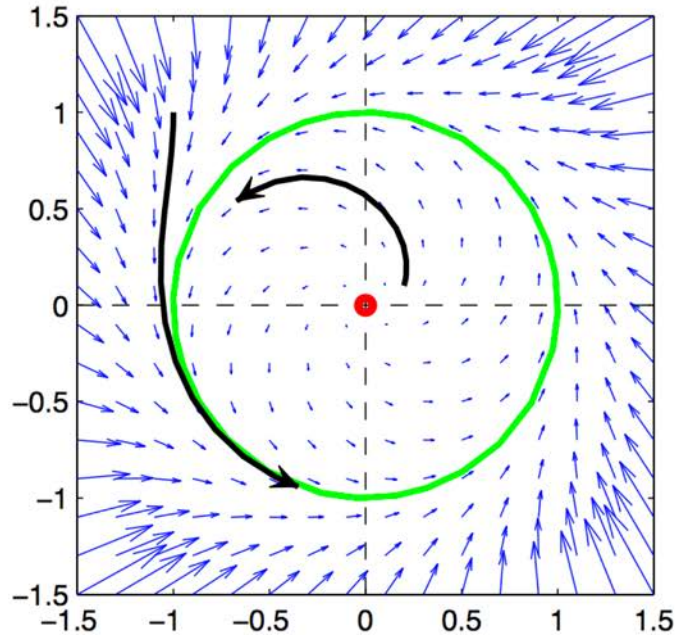
$$\dot{r} = -r^3 + r$$
$$\dot{\varphi} = 1.$$

$$\dot{r} = -r^3 + r + 1/2 \, \sin(3\,\theta) \sin(r)$$
$$\dot{\theta} = 1 + 1/2 \, \sin(4\,r)$$

- We added some fancy (random) extra terms to the RHS of the system equations.

- These extra terms "wobble" the original equations, but do not destroy their basic nature.

$$\dot{r} = -r^3 + r + 1/2 \, \sin(3 \cdot 0.5) \sin(r)$$

original
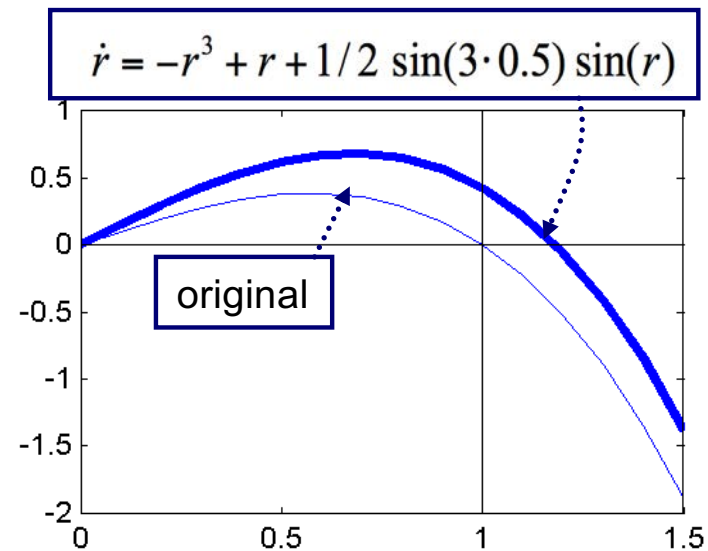
Structural stability
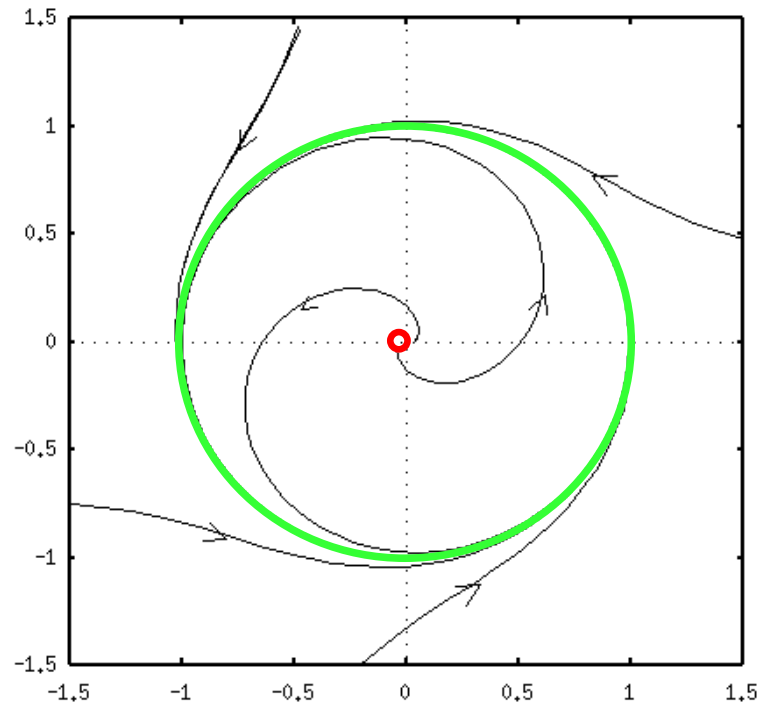
- Two DS are structurally similar if their phase portraits can be continuously transformed into each other.

- Structural similarity is an equivalence relation. It is used to classify dynamical systems.
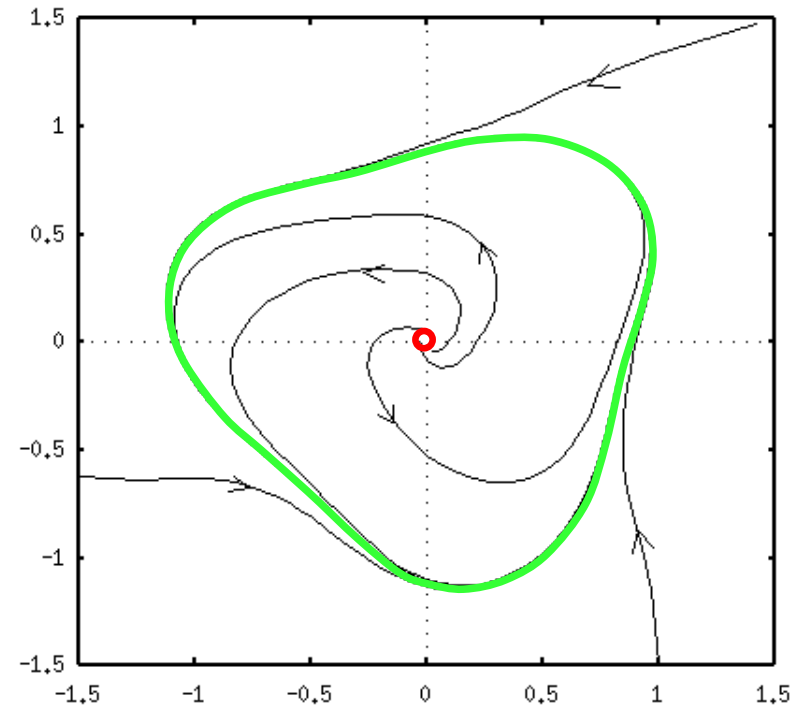
- A DS is structurally stable if small changes to its vector field lead to structurally similar phase portraits.

- Our example system is structurally stable.

Structural instability
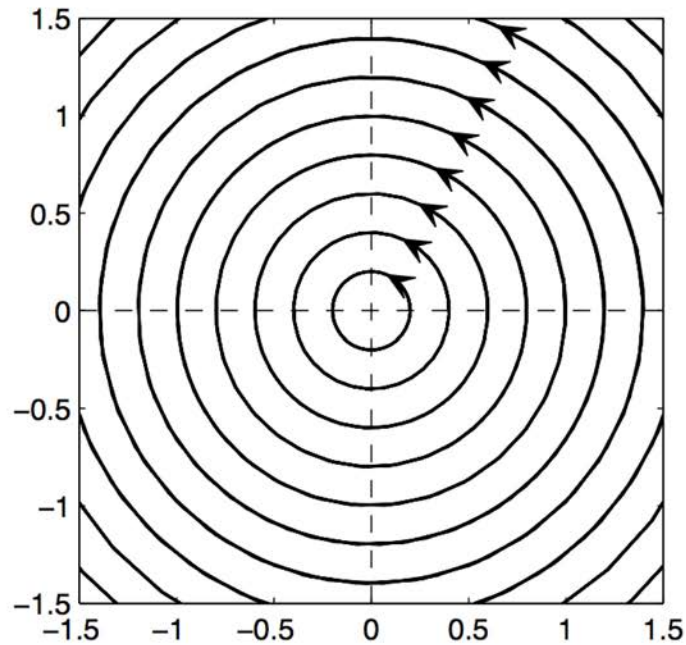
$$\dot{r} = 0 \qquad \dot{\theta} = 1$$
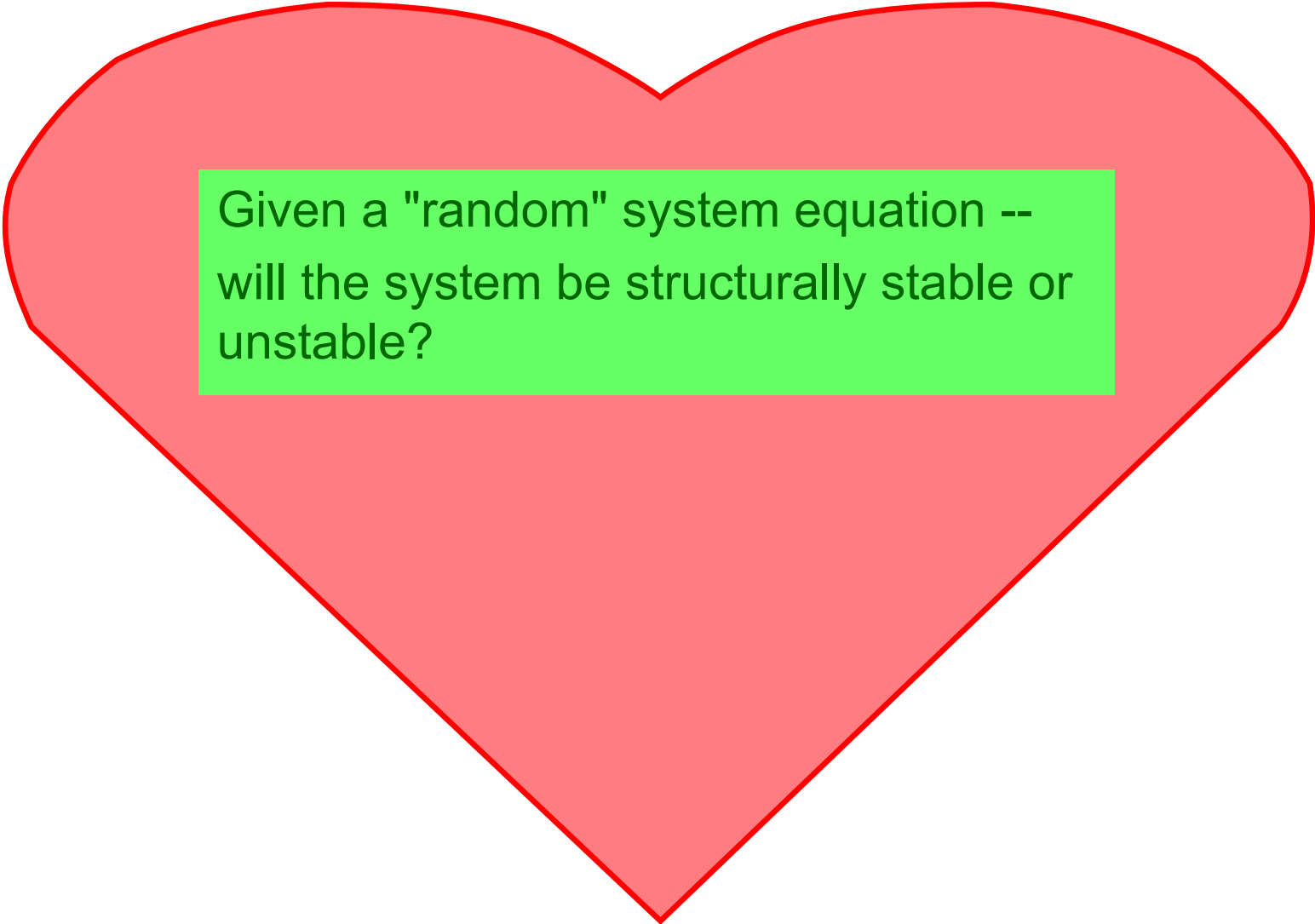
- This system is not structurally stable.
- Arbitrarily small changes in ist equations will make the circles "miss themselves on their return trip" -- giving spiral portraits instead (inbound or outbound)
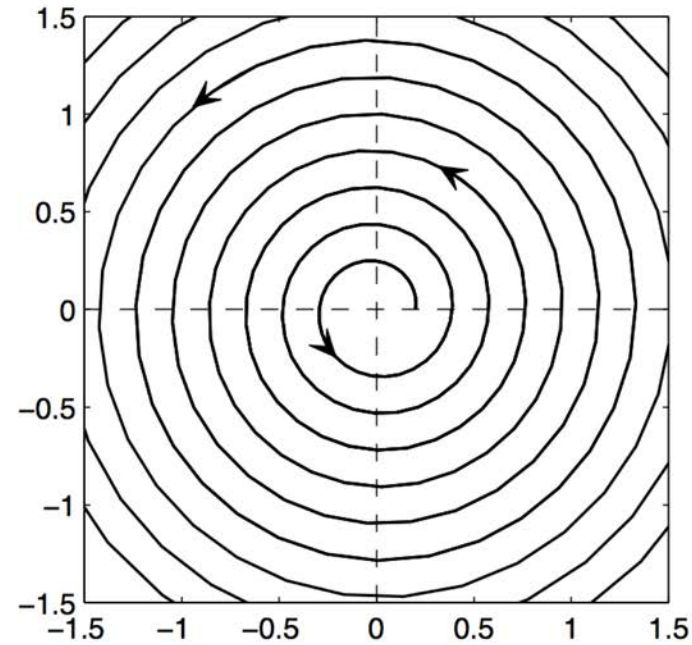
$$\dot{r} = 0.01 \qquad \dot{\theta} = 1$$

- This "spiral" phase portrait is structurally stable.

A question relating to you, maths, and the universe

Given a "random" system equation --
will the system be structurally stable or unstable?

Ich spazierte einsmals im Wald herum meinen eitelen Gedanken Gehör zu geben, da fand ich ein steinern Bildnis liegen in Lebensgröße [...] da fing es an sich zu regen und zu sagen: "Lasse mich mit Frieden, ich bin Baldanders." [...]

 [Er] nahm darauf mein Buch, so ich eben bei mir hatte, und nachdem er sich in einen Schreiber verwandelt, schrieb er mir nachfolgende Worte darein: "Ich bin der Anfang und das End, und gelte an allen Orten. Magst dir selbst einbilden wie es einem jeden Ding ergangen, hernach einen Discurs daraus formirn und davon glauben was der Wahrheit ehnlich ist, so hast du was dein narrischer Vorwitz begehret."

Als er dies geschrieben, wurde er zu einem großen Eichbaum, bald darauf zu einer Sau, geschwind zu einer Bratwurst und unversehens zu einem großen Baurendreck (mit Gunst), er machte sich zu einem schönen Kleewasen, und ehe ich mich versah, zu einem Kuhfladen; item zu einer schönen Blum oder Zweig, zu einem Maulbeerbaum und darauf in einen schönen seidenen Teppich etc. [...] verändert' er sich in einen Vogel, floh schnell davon und ließ mir das Nachsehen.

Grimmelshausen, Simplicius Simplicissimus, sechstes Buch, Kapitel 9

I was once walking around in the forest and listening to my idle thoughts when I found a life-sized stone statue lying on the ground. [...] it began to move by itself and said: "Leave me alone. I'm Soonchanged." [...] Then he took the book which I happened to have with me and, after he had changed himself into a scribe, he wrote the following words in it: "I am the beginning and the end, and I am valid everywhere. [...]" After hed written this, he became a large oak tree, then a sow and then quickly a sausage, and then some peasants dung. The he changed himself into a beautiful meadow of clover and, before I could turn around, into a cow-pie; then he became a beautiful flower or sprout, a mulberry tree and then a beautiful silk rug and so on till he finally changed back into human form [...] Then he changed himself into a bird and flew quickly away. (Grimmelshausen (2012), translation by Monte Adair)

Parameter magic...

$$\dot{r} = -r^3 + a\,r; \quad \dot{\theta} = 1$$

$$a = -1$$
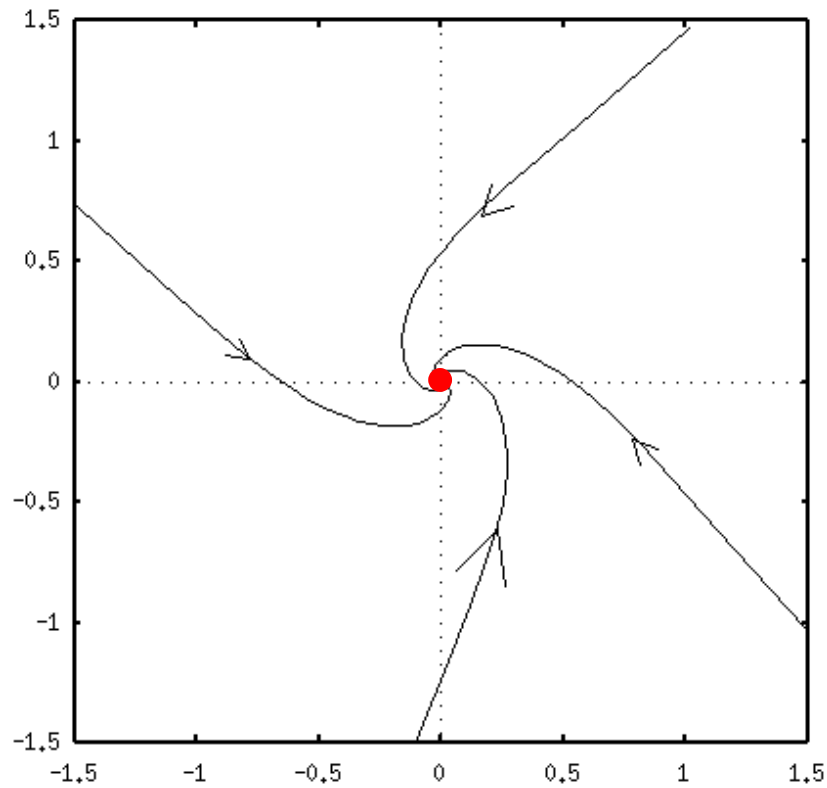
$$\dot{r} = -r^3 - 1r$$

Parameter magic...

$$\dot{r} = -r^3 + a\,r; \quad \dot{\theta} = 1$$

$$a = -0.2$$

$$\dot{r} = -r^3 - 0.2\,r$$

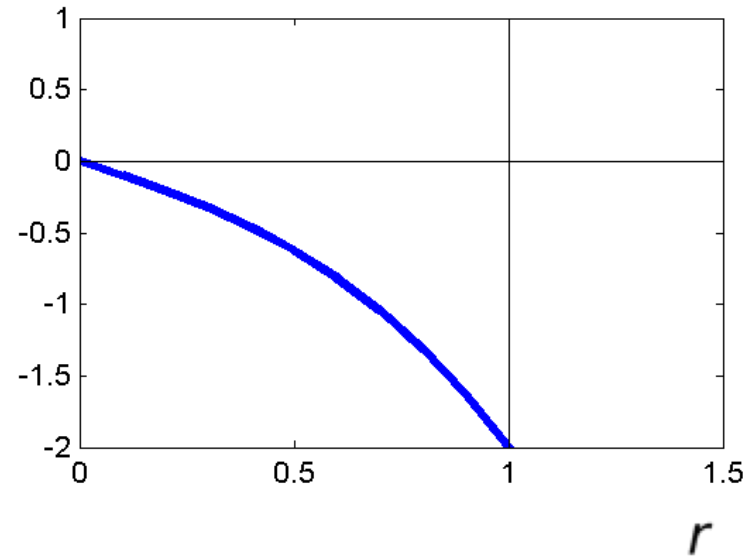# Parameter magic...

$$\dot{r} = -r^3 + a\,r; \quad \dot{\theta} = 1$$

$$a = 0$$
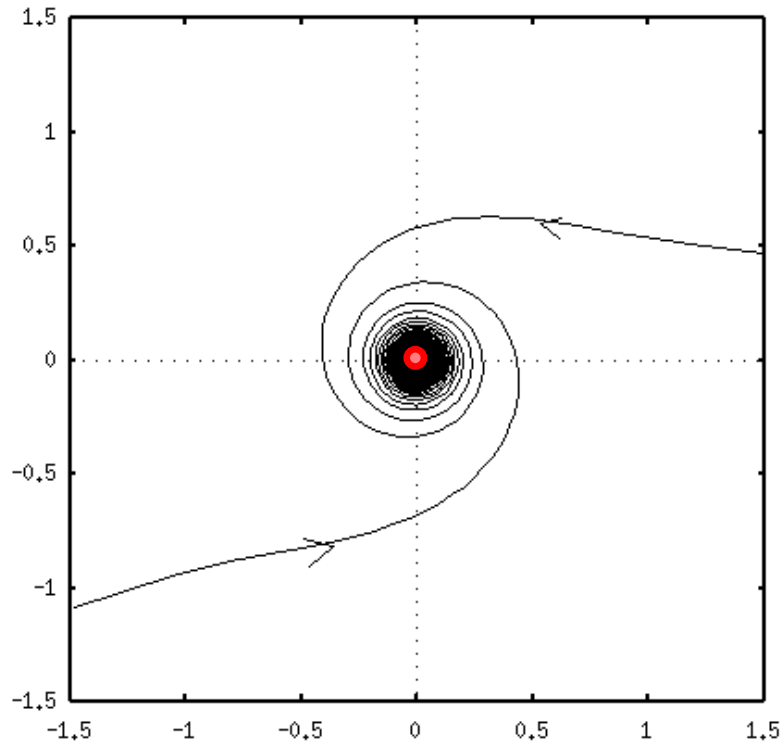
$$\dot{r} = -r^3 - 0\,r$$
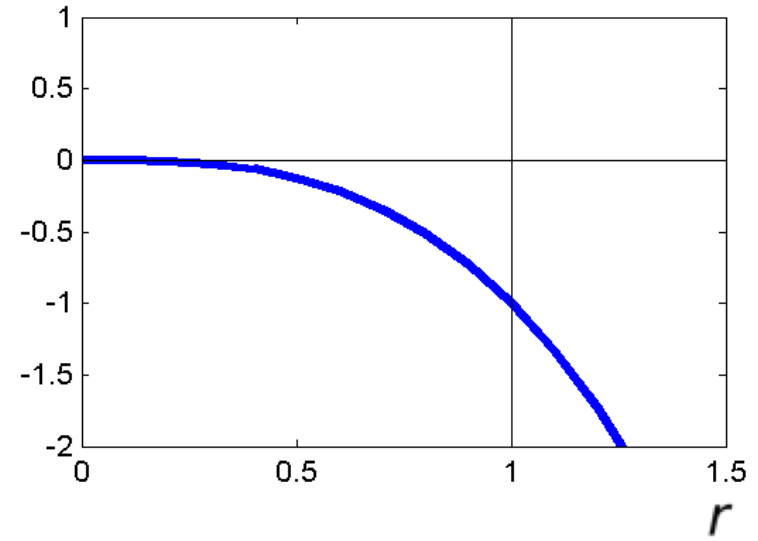
Parameter magic...

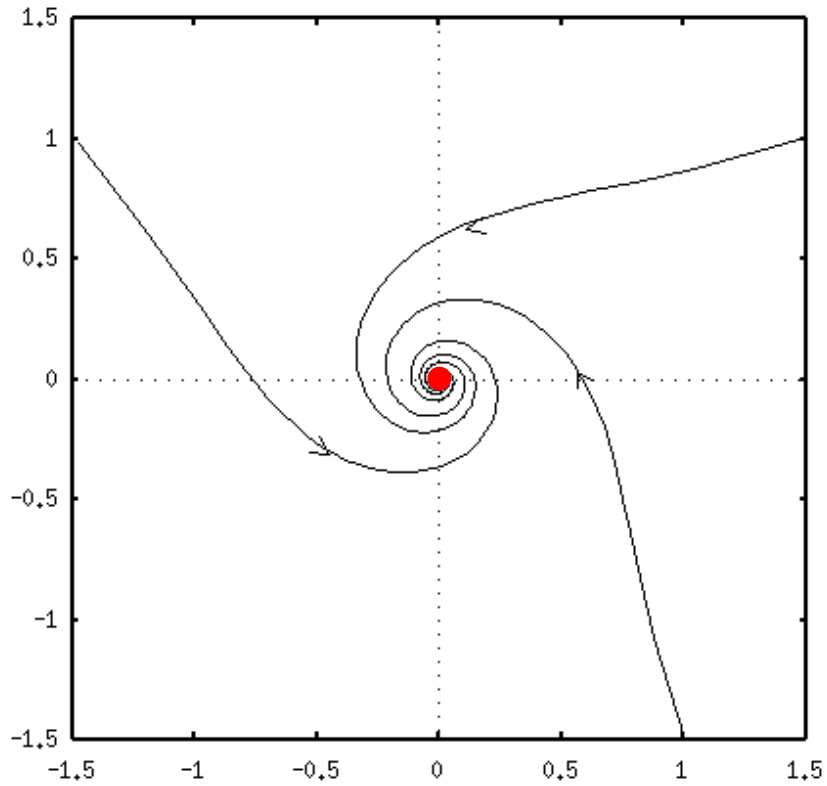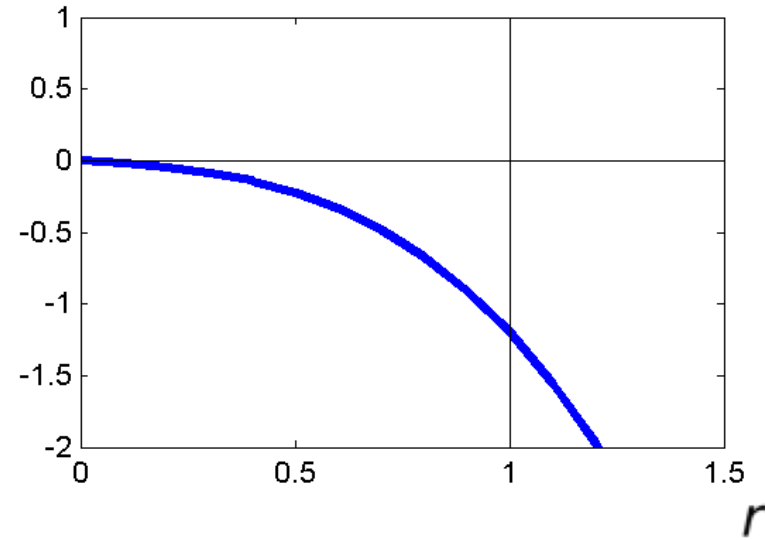$$\dot{r} = -r^3 + a\,r; \quad \dot{\theta} = 1$$

$$a = 0.1$$
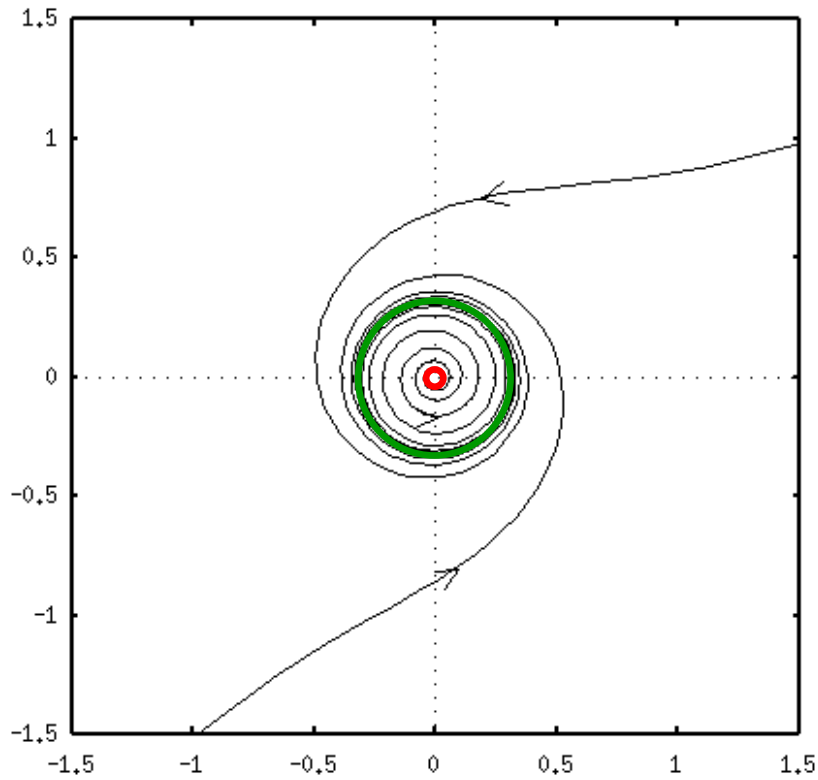
$$\dot{r} = -r^3 + 0.1r$$
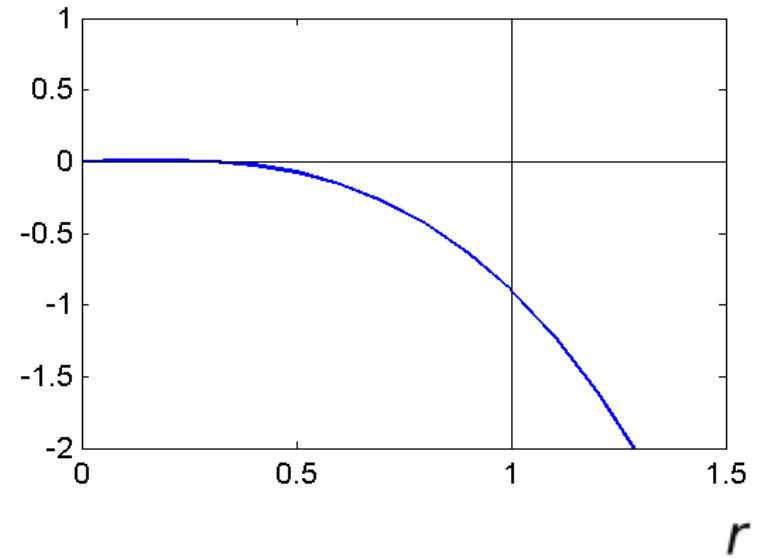
Parameter magic...

$$\dot{r} = -r^3 + a\,r; \quad \dot{\theta} = 1$$

$$a = 1$$
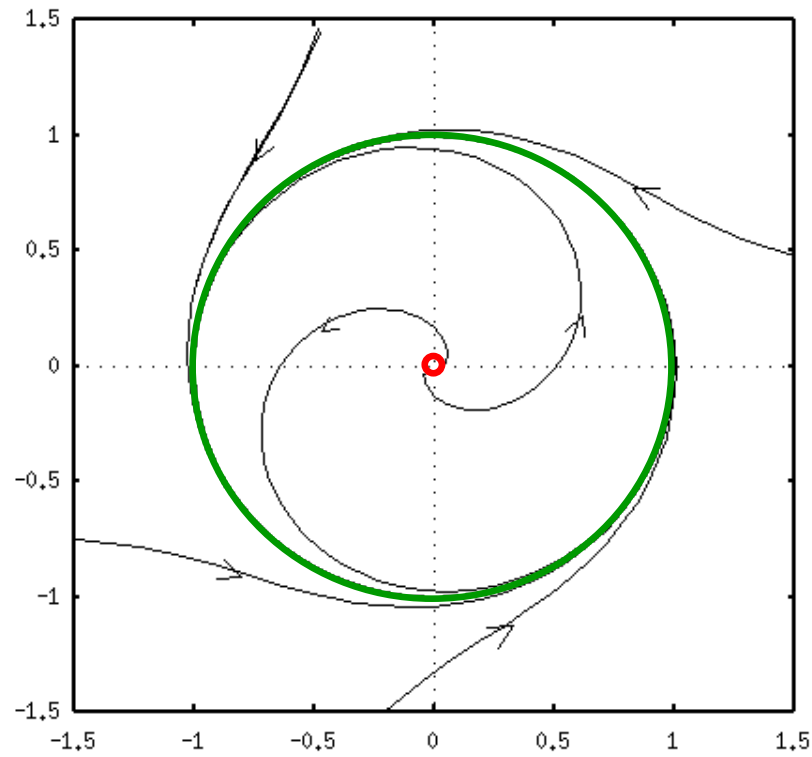
$$\dot{r} = -r^3 + 1r$$
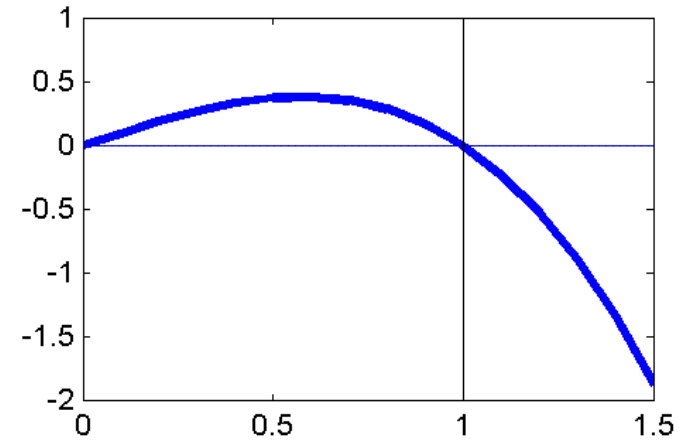
r

It's a Bifurcation.

$$\dot{r} = -r^3 + a\,r; \quad \dot{\theta} = 1$$
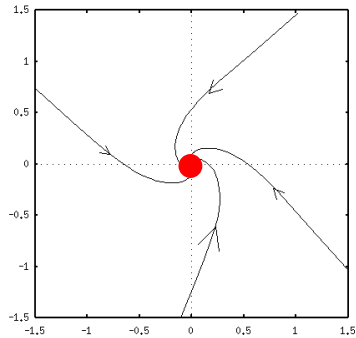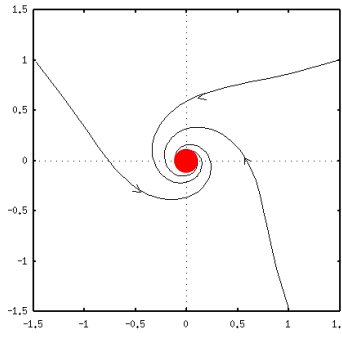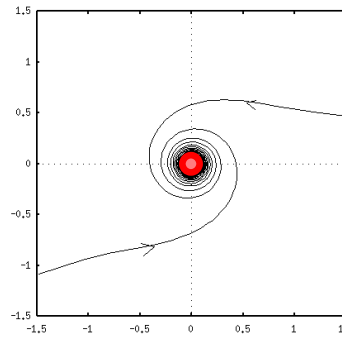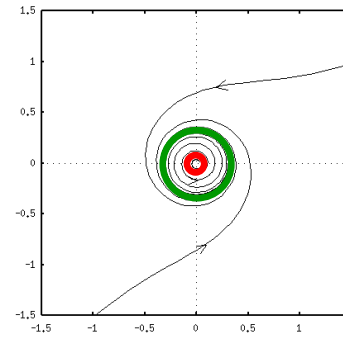
$a = -1$ $a = -0.2$ $a = 0$ $a = 0.1$ $a = 1$
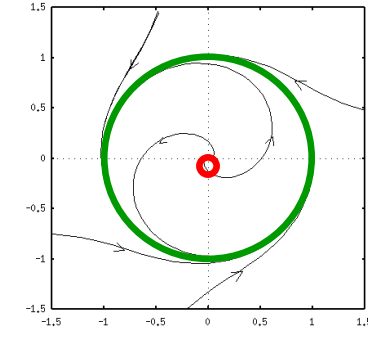
a < 0: Stable fixed point at origin (a point attractor)

Trajectories toward this point have finite length.

System is structurally stable.

a = 0: Stable fixed point, infinite trajectories.

Structurally instable.

a > 0: Repellor at origin, plus a limit cycle attractor.

(reverse) trajectories toward this point have finite length.

System is structurally stable.

Bifurcation: when a control parameter passes through a critical value, the phase portrait changes its nature.

At (and only at) the critical value, the system is structurally instable. "Left" and "right" of this value, the phase portrait is structurally stable.

A zoo of bifurcations

Name	"Left" behaviour	Behaviour at critical value	"Right" behaviour
Supercritical Hopf bifurcation	1 point attractor	1 point attractor with "critical slowdown"	1 point repellor, 1 cyclic attractor (whose amplitude grows from zero)
Subcritical Hopf bifurcation	1 point attractor	1 point attractor, 1 cyclic trajectory which is attracting "from left" and repelling "to the right"	1 point attractor, 1 cyclic attractor (which appears out of the blue with nonzero amplitude), 1 cyclic repellor
Pitchfork bifurcation	1 point attractor (or one cyclic attractor)	1 point attractor with "critical slowdown"	2 point attractors (or two cyclic attractors), 1 repellor (or cyclic repellor)
Saddle-node bifurcation	[no attractor or repellor or other fixed point]	1 fixed point, a saddle node, neither attractor nor repellor	1 point attractor, 1 point repellor

... and there are many more ...

Literature / Resources

The ABC (attractors, bifurcations, chaos) aspects of DS theorie(s) are amply covered in many books and online tutorials.

A (in fact, *the*) web portal with pointers is https://dsweb.siam.org/ .

My best-liked standard textbook is

Strogatz, S.H., Nonlinear Dynamics and Chaos. Addison Wesley 1994 (latest reprint: 2015)

My absolute top-super-favourite book is

Abraham, R.H., Shaw, C.D., Dynamics: The Geometry of Behavior. Addison-Wesley, 1992 (http://www.aerialpress.com/dyn4cd.html offers a recent revision as e-book for 30 USD, used [expensive] paper copies available at Amazon)

This book is a miracle in didactics. The first author is a pioneer DS mathematician, the second a graphics artist. The book explains concepts, phenomena and theory of DS, from first steps to fairly advanced topics, without a single equation, yet rigorously – all done by brilliantly insightful hand-drawn graphics – I have never seen something alike.

Phase transitions

- Bifurcations manifest themselves as a sudden qualitative change of behavior when a control parameter passes a critical value.

- Similarly (?), physical systems may undergo qualitative changes called phase transitions (PT) when control parameters pass critical values

- PTs are defined in a context of statistical thermodynamics, not dynamical systems.

- PTs are defined for statistical ensembles (like water molecules) in the infinite-size limit – for infinite-dimensional stochastic systems. Bifurcations are (originally) defined for low-dimensional deterministic systems.

- classical Definition: a PT occurs when (a derivative of) the free energy of a thermodynamical system passes through a singularity.

Phase transition diagram of water

- PTs are not intrinsically related to bifurcations.

- When macroscopic measurables of a statistical thermodynamical system is described in a low-dim mean-field model, PTs may become modeled by bifurcations.

http://schools.birdville.k12.tx.us/cms/lib2/TX01000797/Centricity/Domain/912/ChemLessons/
Lessons/Phases%20and%20Changes/Phase%20Diagrams.htm

Literature / resource

A didactic website to explore phase transitions with beautifully interactive visualizations is

http://www.ibiblio.org/e-notes/Perc/contents.htm

A brief visit to chaos – the Lorenz attractor

A continuous-time DS in 3 dimensions. System equations:

$$\dot{x} = \sigma(y - x),$$
$$\dot{y} = x(\varrho - z) - y,$$
$$\dot{z} = xy - \beta z.$$

Classical parameters:

$$\sigma = 10, \varrho = 8/3, \beta = 28.$$

A long trajectory
on the attractor
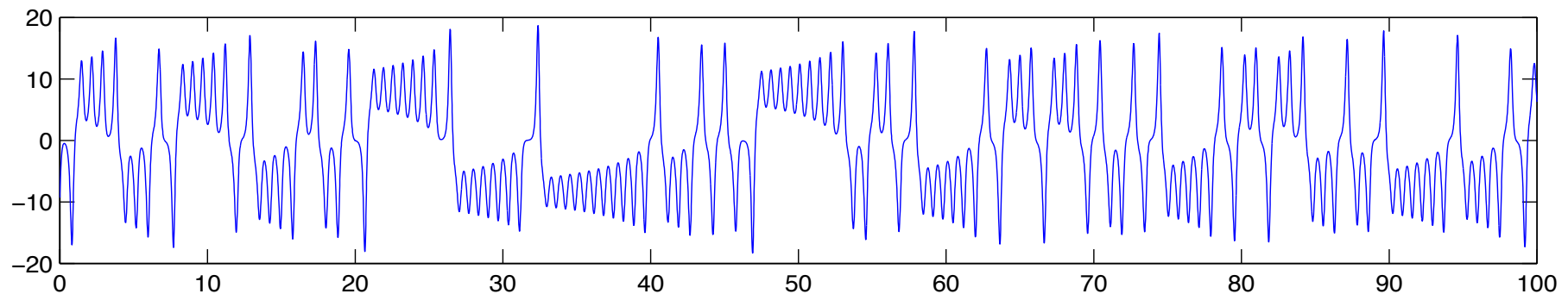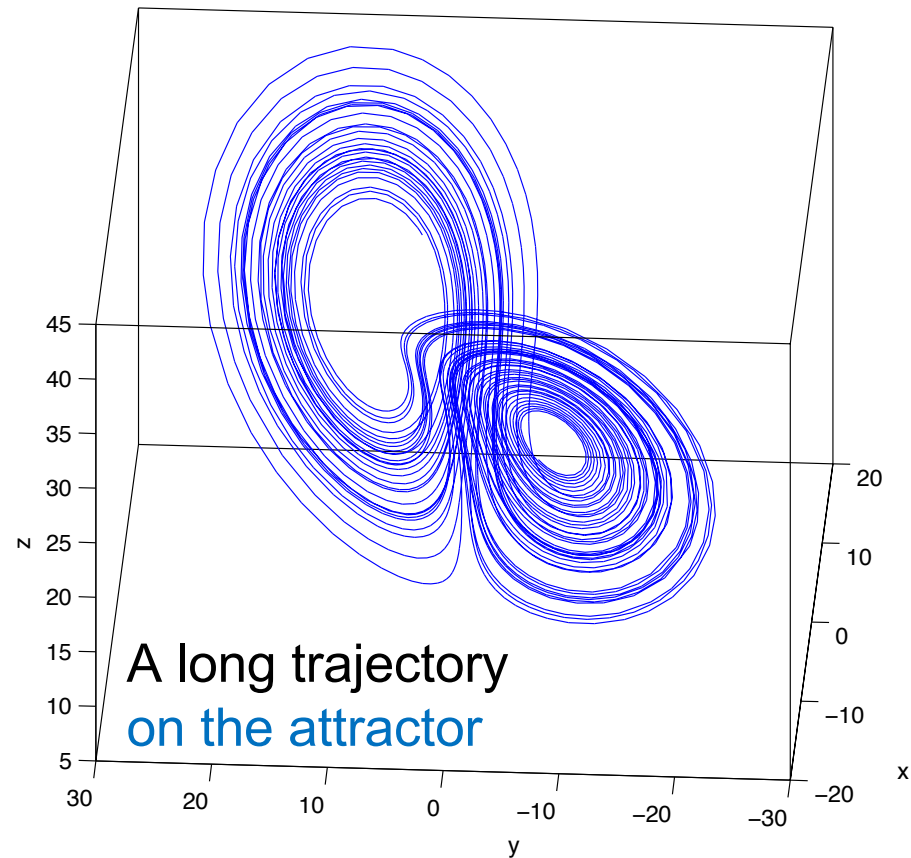
The $x(t)$ component

The hallmark of chaos:

exponential divergence of nearby trajectories
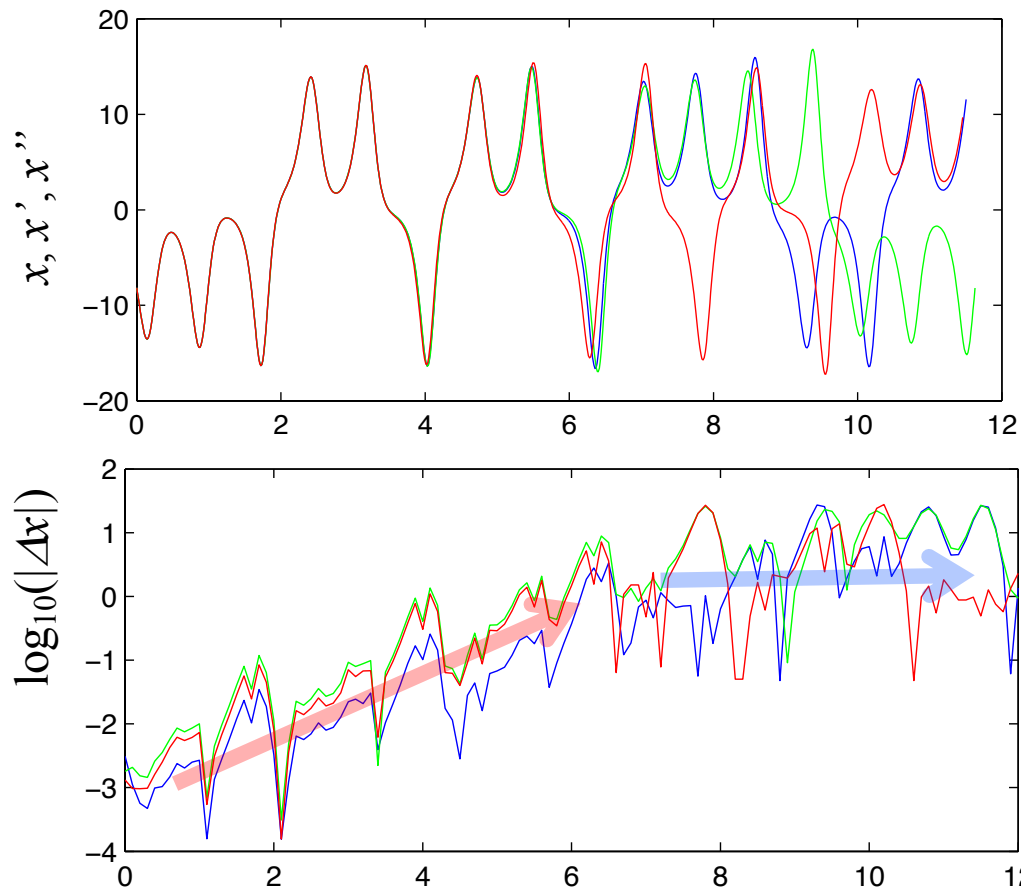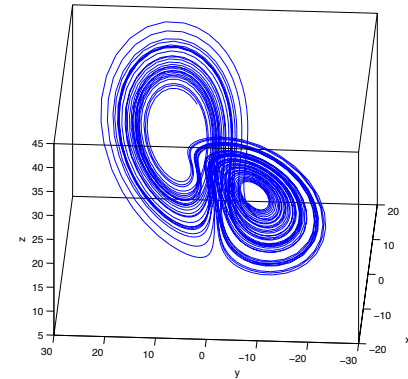
Three $x(t)$ traces, started at distances 1/10,000 of attractor diameter from each other

$\log_{10}$ distances of trajectories: exponential separation until average distance = order of attractor size

Miscellaneous notes

- Attractor is a fractal subset of state space

- Chaos often arises in "highly self-exciting" DS

- For instance, in recurrent neural networks with strong self-feedback

- In nature: turbulent flows (atmosphere, ocean circulations, airflow, plasmas)

- Butterfly effect:

    - **Given**: two almost identical earth atmosphere simulations A and B

    - Only **difference**: in simulation B, a butterfly in China flaps its wings once

    - **Result**: after some weeks, a tornado in Günne in A, calm sunshine in B

- Though deterministic, chaotic DS are practically unpredictable because initial state cannot be known with 100% precision
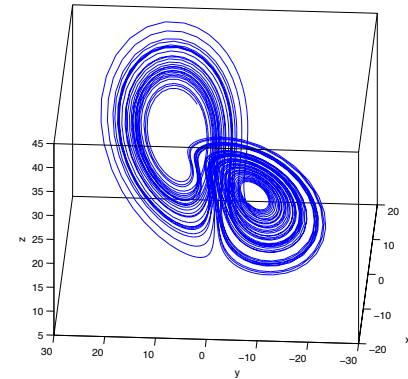
Attractors mathematically defined

Topological dynamics

- Dynamical systems relevant for neuroscience modeling come with a variety of state spaces, e.g.

  - $\mathbb{R}^n$,
  - manifolds in $\mathbb{R}^n$,
  - function spaces,
  - σ-fields (aka σ-algebras),
  - sets of words or symbol sequences.

- Desirable: a general theoretical view on all of these

- Specifically: define general concepts (i.e., get deeper insight) of attraction, bifurcations

- Unifying perspective: investigate dynamics on topological spaces

---

Definition. Let X be a set, O $\subseteq \mathrm{Pot}(X)$ a collection of subsets, called the open sets. Then $(X,$ O$)$ is a topological space if

(i) $\emptyset, X \in$ O,

(ii)   O is closed under arbitrary unions,

(iii)   O is closed under finite intersections

Definition. Let $A \subseteq N \subseteq X$. Then N is a neighborhood of A if $A \subseteq U \subseteq N$ for some $U \in$ O.

Attractors defined in topological dynamics

Definition. Given: topological state space X, map $T: X \rightarrow X$. A set $A \subseteq X$ is an attractor of T if

1. A is invariant under T, i.e. $T(A) \subseteq A$

2. There is a neighborhood B of A, called the basin of attraction, where $B(A)$ is the set of all $b \in X$ satisfying

   for any open neighborhood U of A the T-iterates of b are ultimately confined to U, i.e.

   $$\exists k \, \forall n > k : T^n(b) \in U$$

Notes.

- Some authors call this an attracting set, and for an attractor in addition require minimality or that some T-orbit is dense in A.

- Several subtly non-equivalent definitions. Watch out for the small print!
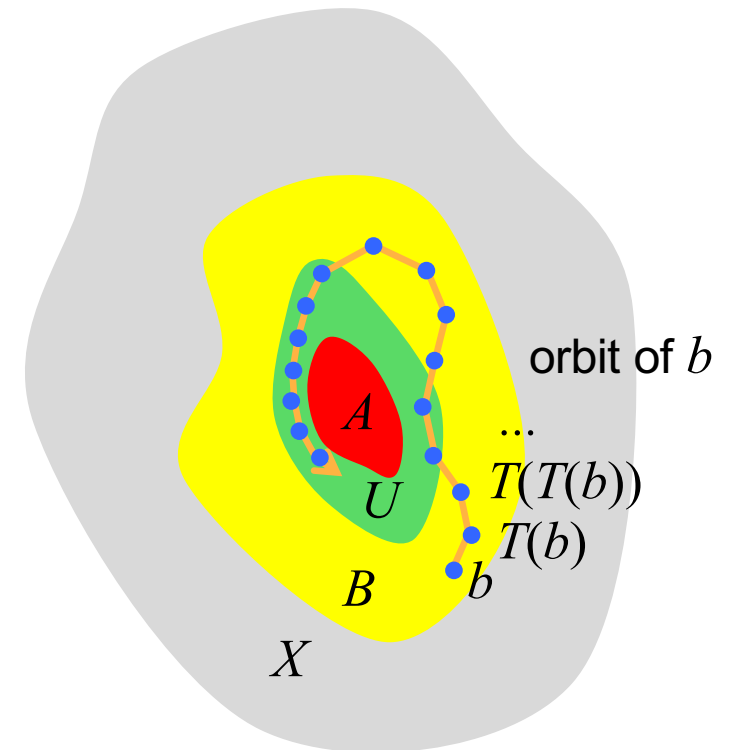
Example: cyclic attractors and their basins in a 16-cell CA

see detail

garden-of-Eden states

attractor cycle

transient trees and sub-trees
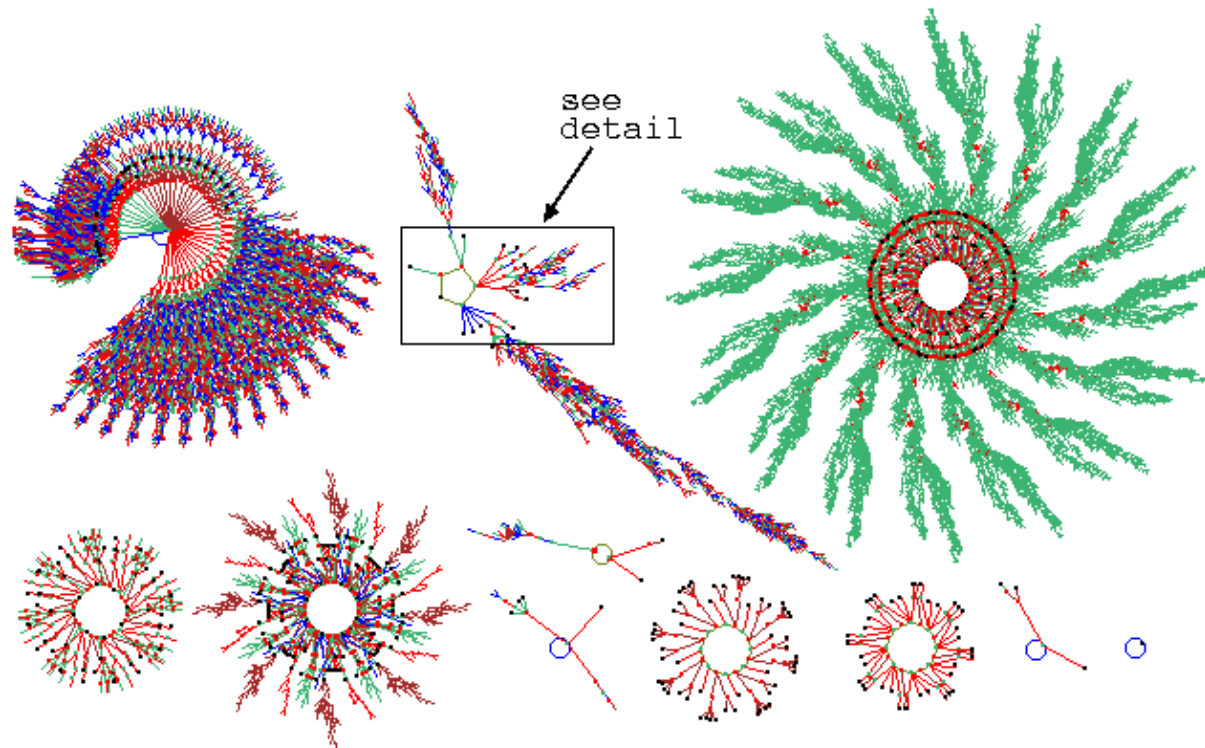
- State space X: binary 16-bit patterns

- CA transition rule yields $T: X \rightarrow X$

- discrete topology $O = \mathrm{Pot}(X)$

http://uncomp.uwe.ac.uk/wuensche/gallery/ddlab_gallery.html
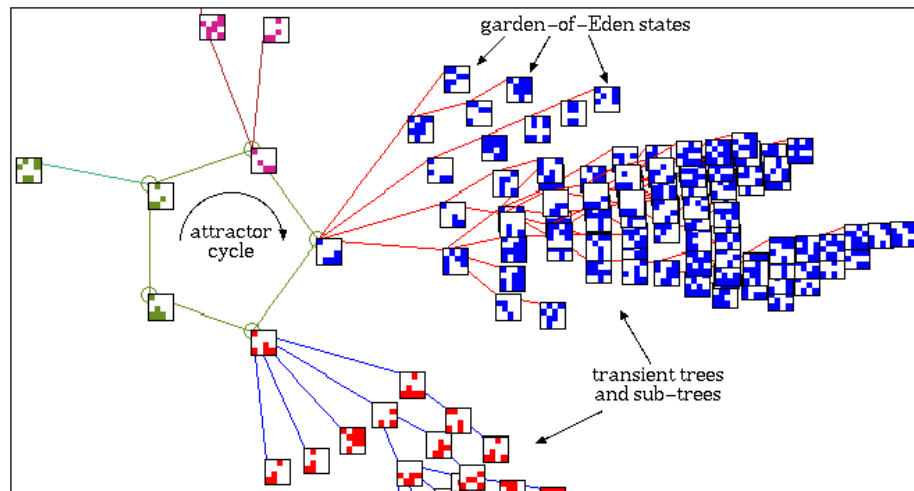
Example: a point attractor in a continuous 2-dim DS

fixed point attractor

basin of attraction
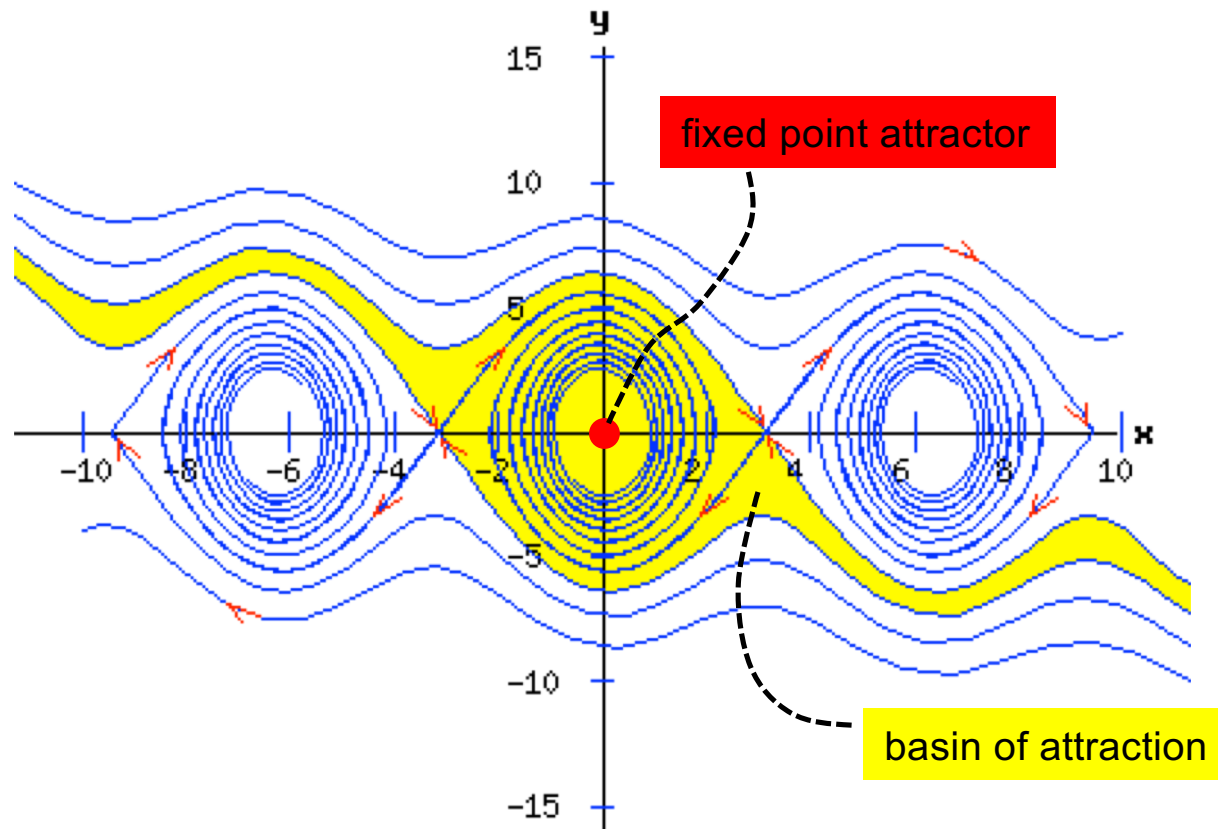
- State space X: $\mathbb{R}^2$

- continuous dynamics generated by

$$\dot{x} = y$$

$$\dot{y} = -9\sin(x) - y/5$$

- standard $\mathbb{R}^2$ topology (induced by Euclidean metric)

http://www.egwald.ca/nonlineardynamics/twodimensionaldynamics.php

Literature

Very accessible, yet comprehensive and rigorous (in other words, just very well-written):

Ethan Akin: Topological Dynamics: A Survey. Online manuscript, Dpt. of Mathematics, City College New York, 2007. Find it at
http://math.sci.ccny.cuny.edu/people?name=Ethan_Akin.

(Also appeared as chapter "Topological Dynamics" in the Encyclopedia of Complexity and Systems Science (R. A. Meyers, ed.), Springer Verlag 2008)

Attractors and symbols

The riddle

Neurons fire.

Humans think.

I think, therefore I am
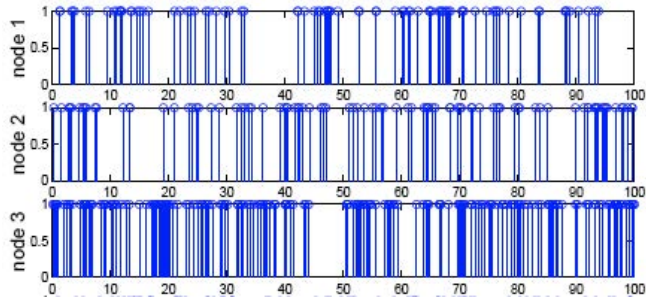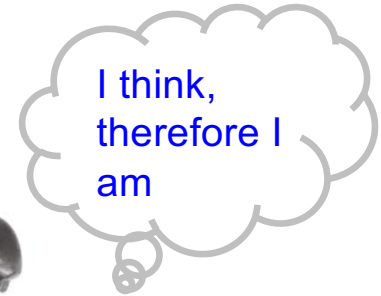
The golden riddle of neural-symbolic integration

- Thinking uses concepts, words, symbols -- stable, identifiable "tokens"

- What are the neural correlates?

- One natural modelling approach: symbols = attractors in neural dynamics

Other modelling approaches

Other candidates for neural correlates of symbols:

- discrete regions in neural state space

- individual neurons or areas (space coding)

- dimensions / subspaces in neural state space (the core of most artificial neural networks trained for classification)

- saddle nodes (theory of homoclinic cycles)

- conceptors

Literature

Overview articles (from a somewhat specific perspective) on neural-symbolic integration:

Besold, T. et al (2017): Neural-Symbolic Learning and Reasoning: A Survey and Interpretation https://arxiv.org/pdf/1711.03902

L. C. Lamb. The grand challenges and myths of neural-symbolic computation. In L. De Raedt, B. Hammer, P. Hitzler, and W. Maass, editors, Recurrent Neural Networks - Models, Capacities, and Applications, number 08041 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008 http://drops.dagstuhl.de/opus/volltexte/2008/1423

Symbols as regions in neural state space, e.g.:

Tino, P. and Cernansky, M. and Benuskova, L.: Markovian Architectural Bias of Recurrent Neural Networks. IEEE Trans. on Neural Networks 15(1), 2004, 6-15

Symbols as saddle nodes:

Gros, C.: Cognitive computation with autonomously active neural networks: an emerging field. Cognitive Computation 1, 2009, 77-90

Symbols as conceptors:

H. Jaeger (2014): Conceptors: an easy introduction. http://arxiv.org/abs/1406.2671

Symbols as attractors

Pros

- accounts naturally for temporal stability of using "symbols" in neural processing

- *the* classical model of a neural memory for discrete items: Hopfield networks.

    - Here point attractors = learnt concepts

Con

- inherent conflict: attractors by definition captivate neural trajectory forever, while thinking transits from concept to concept

Attempts to resolve the inherent conflict:

- neural noise (kicks trajectory out of attractor)

- generalized / modified attractor concepts:

    - chaotic itinerancy

    - attractor relics, attractor ruins, or attractor ghosts;

    - transient attractors

    - unstable attractors

    - high-dimensional attractors (initially named partial attractors)

    - attractor landscapes

- Cautious authors speak of attractor-like phenomena

7. Non-Autonomous Dynamical Systems

Context

- ABC theory was developed toward its current striking beauty by pure mathematicians working on autonomous (input-free, deterministic, stationary) systems.
- But: real-life neural dynamics is input-driven, stochastic, non-stationary: it is non-autonomous
- The mathematical theory of non-autonomous DS is young, much more involved than autonomous DS theory, very incomplete

Personal opinion: many guiding metaphors borrowed from autonomous DS theory are misguiding.

Specifically, this holds for attractor concepts.

One reason for the slow progress in neuro-symbolic integration programme: inavailability / inaccessibility of a suitable non-autonomous theory of attractors

Non-autonomous systems, general definition

A non-autonomous (discrete-time) DS is given by an update equation

$$\mathbf{x}(n+1) = T_n(\mathbf{x}(n))$$

- The state update law is now time-dependent itself: T_n instead of T
- Time-dependency can arise from various sources, e.g.
  - input: $T_n(\mathbf{x}(n)) = T(\mathbf{x}(n), \mathbf{u}(n))$,
  - random perturbations: $T_n(\mathbf{x}(n)) = T(\mathbf{x}(n)) + v(n)$,
  - nonstationarity: T_n itself is time-varying
- General theory of non-autonomous dynamical systems (NDS) abstracts away from specific sources of non-autonomy

Example of a non-autonomous system: neural working memory

- a recurrent neural network was trained to predict next symbol in a graphical "video" input stream
- symbol sequence structured by nested curly brackets { { } { { } } }
- different bracket level = different grammar for symbol sequence inside the bracket pair
- to solve task, network must keep track of nesting levels

- each nesting level was "locked into" by bistable memory neurons y
- network had to remain "temporarily stably" locked until next bracket was processed
- modeling intuition: nesting levels ~ attractor states
- due to ongoing input, this is a non-autonomous system and classical attractor notion is not applicable

1200 neuron reservoir

trained
fixed

y_1

y_2

payload task: predict next symbol

z_1 ... z_{65}

WM task: switch according to bracket level (2/6 shown)

switching states of memory units

Literature

The neural working memory is documented in

R. Pascanu, H. Jaeger (2011): **A Neurodynamical Model for Working Memory.** Neural Networks 24(2), 199-207

https://www.ai.rug.nl/minds/uploads/2321_PascanuJaeger10.pdf

This article also contains pointers to a diversity of mathematical approaches for non-autonomous attractors. However, I became aware (thanks to Manjunath Gandhi) of the existing general topological theory of non-autonomous systems only after writing this article, so ignore the attempts for an *ab initio* formalization made in that article.
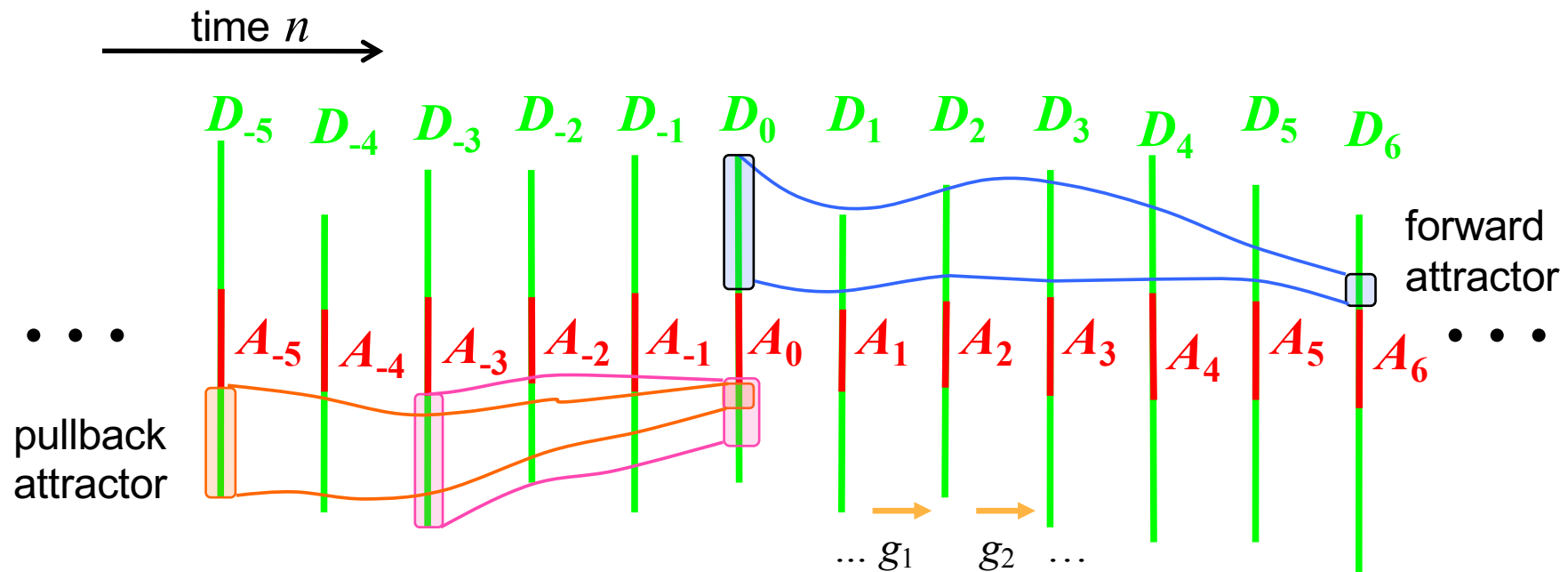
Specific assumptions

We consider

- an individual input sequence $(\mathbf{u}_n)_{n \in \mathbb{Z}}$, $\mathbf{u}_n \in U$;
- a system update equation of the kind

$$\mathbf{x}_{n+1} = f(\mathbf{x}_n, \mathbf{u}_n) =: g_n(\mathbf{x}_n), \ \mathbf{x}_n \in X,$$

where X is metric and compact, and f is uniformly continuous or U is compact.

These assumptions hold for input-driven RNNs (and many real-world systems).

Time-dependent attractors

- consider a time-varying set $(A_n)_{n \in Z}$ that is g-invariant: $g_n(A_n) = A_{n+1}$

- [simplified] such (A_n) is a pullback attractor if there are neighborhoods D_n of A_n such that

$$\forall n : \lim_{k \to \infty} dist(g_{n-1} \circ \cdots \circ g_{n-k}(D_{n-k}), A_n) = 0$$

- [simplified] (A_n) is a forward attractor if

$$\forall n : \lim_{k \to \infty} dist(g_{n+k-1} \circ \cdots \circ g_n(D_n), A_{n+k}) = 0$$

- if an attractor is both forward and pullback, it is a uniform attractor.

Time-independent attractors

Motivation. In the neural working memory example, network states seem to be stably confined in a certain region while operating within a particular parenthesis nesting level. This "attracting set" is time-independent.

Approach (highly simplified)

- graph of each g_n is a set of pairs $\{(\mathbf{x}, g_n(\mathbf{x}))\}$.

- consider the set G of all accumulation points $\{(\mathbf{y}, \mathbf{y}')\}$ where $(\mathbf{y}, \mathbf{y}') \in G$ iff there is a subsequence g_k of the g_n such that $(\mathbf{y}, \mathbf{y}') = \lim_{k \to \infty} (\mathbf{x}, g_k(\mathbf{x}))$.

- G is a time-independent relation.

- Consider G as graph of a time-independent multi-valued function g.

- Define attractors w.r.t. this g: base attractors.
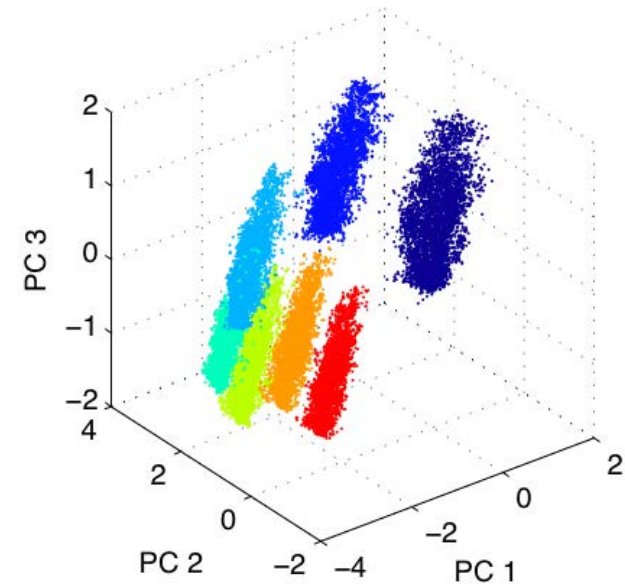
state space regions visited by working memory network when locked in different memory modes.

Non-autonomous dynamics and attractors, comments

- Research started seriously only in the 1990-ies, in pure maths and fragmented application communities

- A diversity of results in specific contexts, e.g.
  - input is from a known stochastic process
  - symbol dynamics

- General theory in the spirit of topological dynamics is very young, not fully ready for end-user adoption

- Not yet available: generally agreed definition of attractors

- Attractivity phenomena are quite different from the autonomous case, new intuitions needed

- Example of novel phenomenon: shovel bifurcation
  - As control parameter passes critical value, a continuous family of "entire solutions" (roughly corresponding to fixed points in autonomous systems) is created

Carreer hint: You are young, healthy, ambitious, mathy?
Make non-autonomous DS your research theme!

Poetzsche 2011
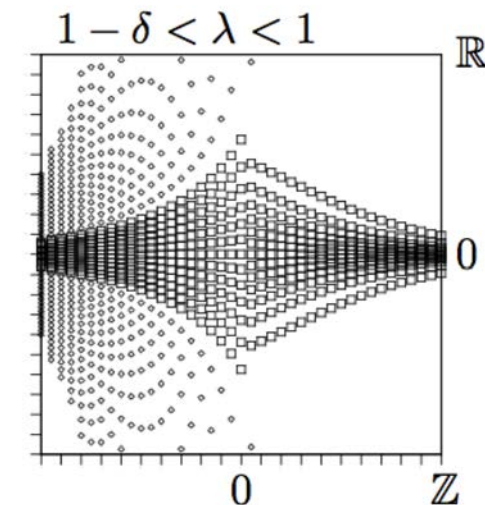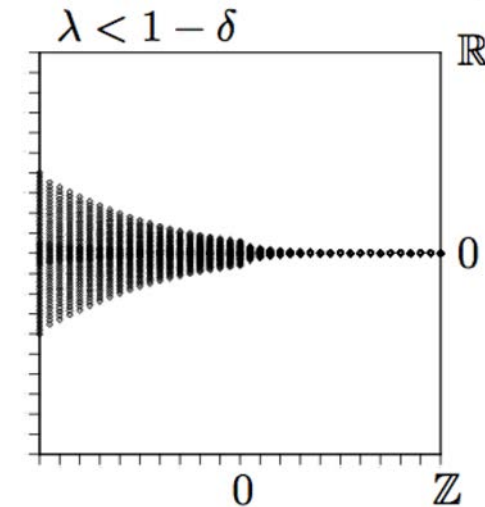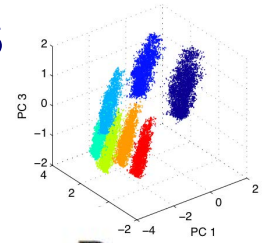
shovel bifurcation

Take-homes

- Everything that *is*, is a dynamical system

- Can't offer you a single universal theory of DS – it can't exist unless there would be a single universal theory of everything

- Across the diversity of DS theories, formalisms and models, some commonly investigated aspects:
    - attractors, bifurcations, chaos (and some more)
    - stability and robustness
    - modeling hierarchies from simplified/abstracted/discrete/symbolic to detailed/precise/numerical
    - qualitative vs. quantitative analyses

- Before you start modeling, step back and contemplate what aspects of dynamics, exactly, you want to capture
    - don't jump to ODEs too quickly

- Deep riddles and steep challenges remain:
    - „grammatical" structure in time series data, structure-forming dynamics
    - neuro-symbolic integration
        - reconstructing „computing" operations in continuous DS: variable binding, memory addressing, information encoding
    - non-autonomous ABC theory

Literature

Current standard textbook:

Kloeden, Peter E., and Martin Rasmussen. Nonautonomous dynamical systems. Volume 176 of Mathematical Surveys and Monographs. *American Mathematical Society, Providence, RI* (2011).

Detailed treatment and some new results to relationships between pullback, forward, and time-independent attractors:

Manjunath, G., Jaeger, H. (2014): The Dynamics of Random Difference Equations is Remodeled by Closed Relations. SIAM Journal on Mathematical Analysis 46(1), 2014, 459-483

https://www.ai.rug.nl/minds/uploads/2499_ManjunathJaeger13a.pdf

Shovel bifurcation introduced in

Poetzsche, C.: Nonautonomous bifurcation of bounded solutions II: A Shovel-Bifurcation pattern. AIMS DCDS-A 31(3), 2011, 941-973 http://wwwu.uni-klu.ac.at/cpoetzsc/Christian_Potzsche_(Publications)/(C)_files/Manuscript_Poetzsche.pdf

Appendix: Themes not Covered in this Tutorial

This tutorial could not cover all of the existing wonderful and powerful and insightful formal methods for describing dynamical systems. Here is a list of what I find the most painful omissions which maybe in future editions I will add:

- general formalism of stochastic processes as sequence of random variables
- ergodicity
- entropy and time arrow
- coupled oscillator systems
- reasoing about time: temporal logic, Allen's time relations
- multiple timescale dynamics