IK 2018

Dynamical Systems: a Navigation Guide

Herbert Jaeger

JACOBS
UNIVERSITY

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 (second third of third session)

JACOBS
UNIVERSITY

# Part 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).
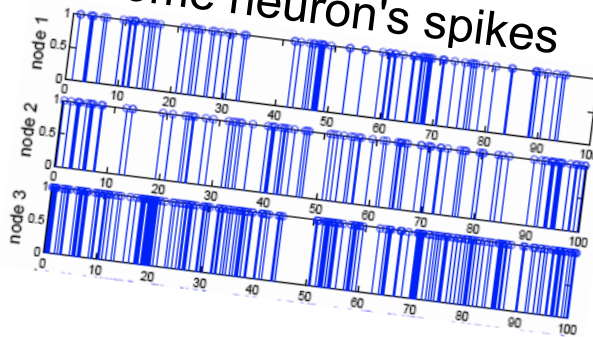
A bit more on the affordable side, at https://dsweb.siam.org/Education

you find an extensive compilation of online tutorials and course materials spanning the whole range of dynamical systems.
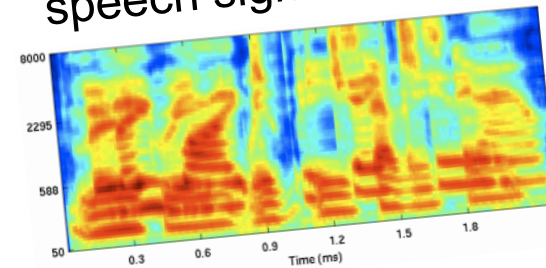
JACOBS UNIVERSITY

Things to describe & analyze
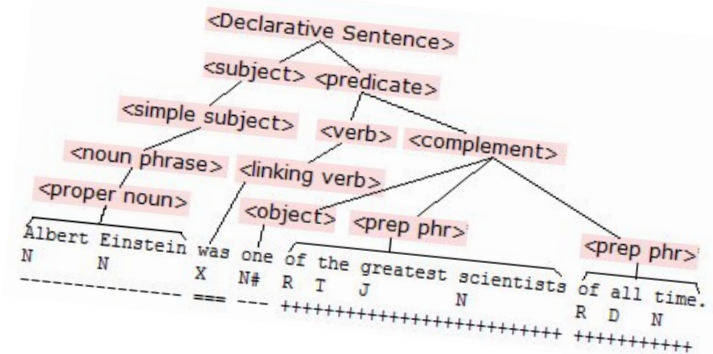
speech signals
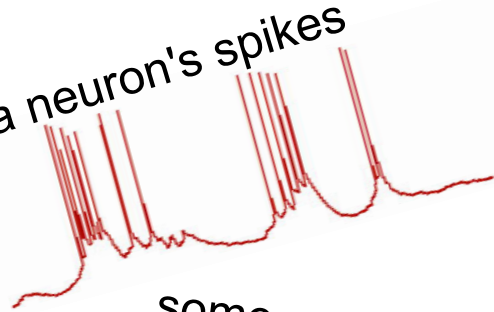
a neuron's spikes
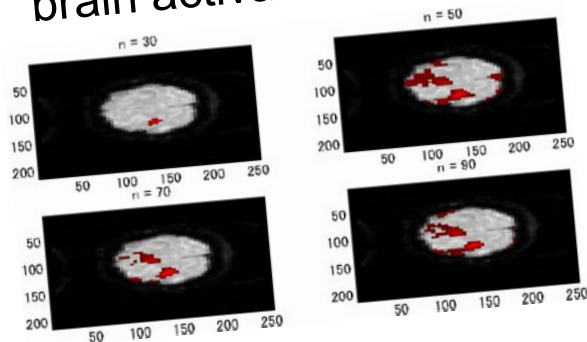

some neuron's spikes

language

brain activation waves

controlled action

JACOBS UNIVERSITY

Forgive me: I start with an a-b-s-t-r-a-c-t picture of modeling dynamics

What is a dynamical system?

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

JACOBS
UNIVERSITY

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
- What you think about you

In sum:

- It is impossible not to *be* a dynamical system.

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

Texts, event and action sequences, DNA, conceptual reasoning

⟷

Numerical

Physiological models, psychometrical measurements, motor control

Non-deterministic

Language competence models, grammatical sequence generation

Deterministic

Electrodynamics, artificial neural networks, mean-field models

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

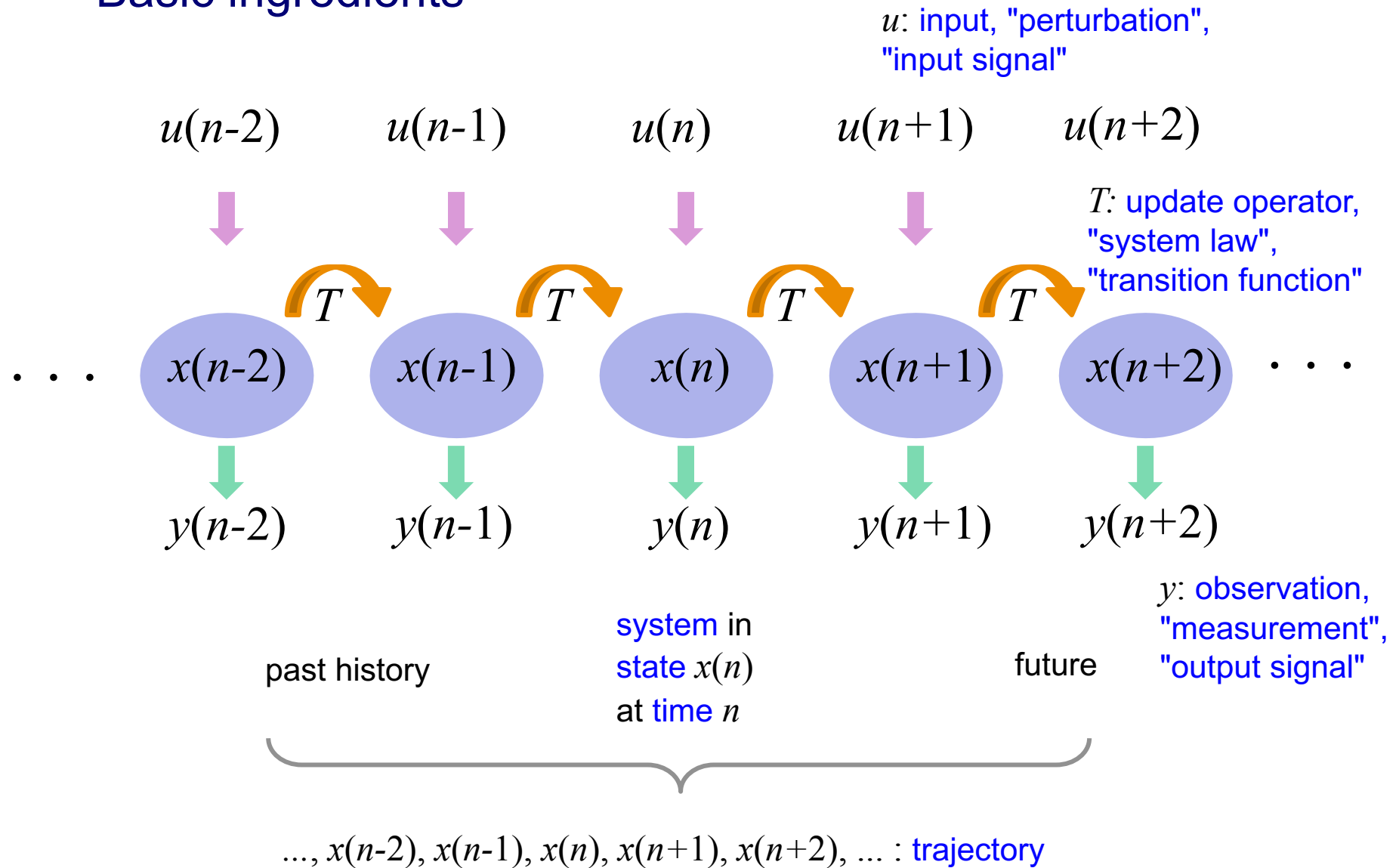- Formalisms: ODEs, PDEs, automata models, dynamical Graphical Models

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
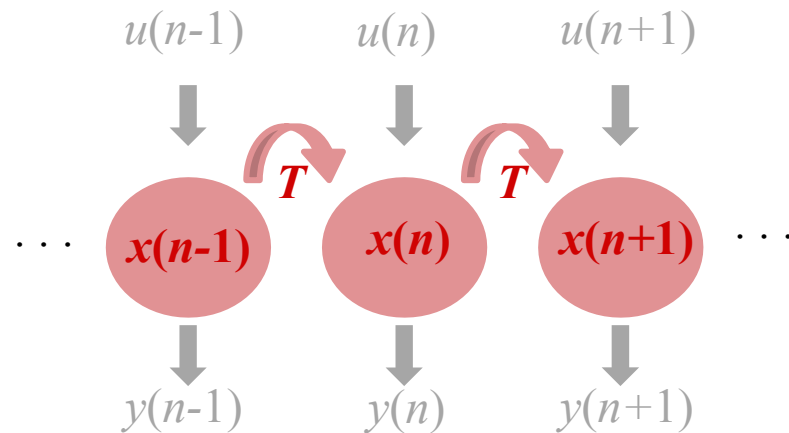
Epistemic modeling (rare)

- Try to emulate information gain processes

- Sometimes adopted by roboticists, agent modeling

- Examples: concept formation processes, belief state modeling

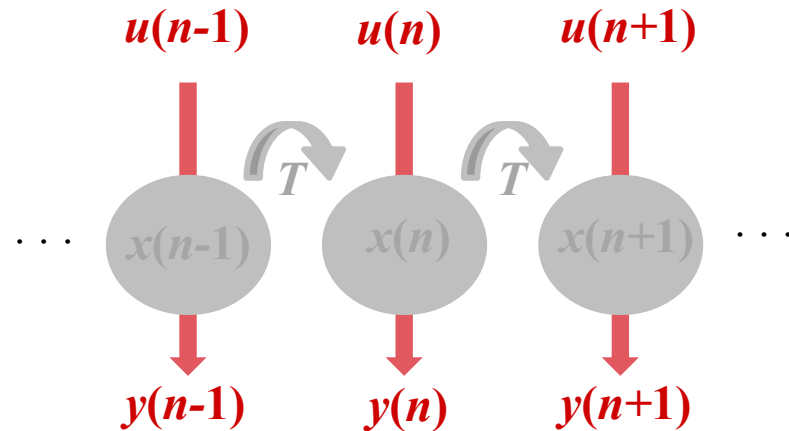- Formalisms: predictive state representations, observable operator models, temporal logics

JACOBS
UNIVERSITY

Basic ingredients

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

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

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

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

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

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

past history   system in state $x(n)$ at time n future

$..., x(n\text{-}2), x(n\text{-}1), x(n), x(n+1), x(n+2), ...$: trajectory

JACOBS UNIVERSITY

The natural science view

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

T T

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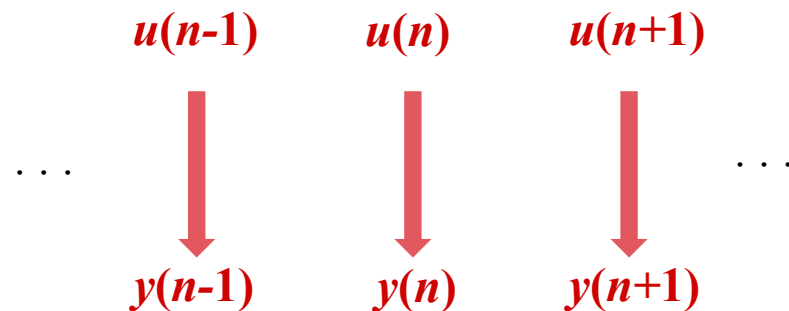
$y(n\text{-}1)$ $y(n)$ $y(n\text{+}1)$

- 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), $x \in \mathbb{R}^n$
- 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: measurable, that is, a vehicle to infer back to state x

JACOBS
UNIVERSITY

The engineering view

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

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

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

"state-based representation"

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

$\cdots$ $\cdots$

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

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/

JACOBS
UNIVERSITY

A first comparison

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

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

$y(n\text{-}1)$ $y(n)$ $y(n\text{+}1)$
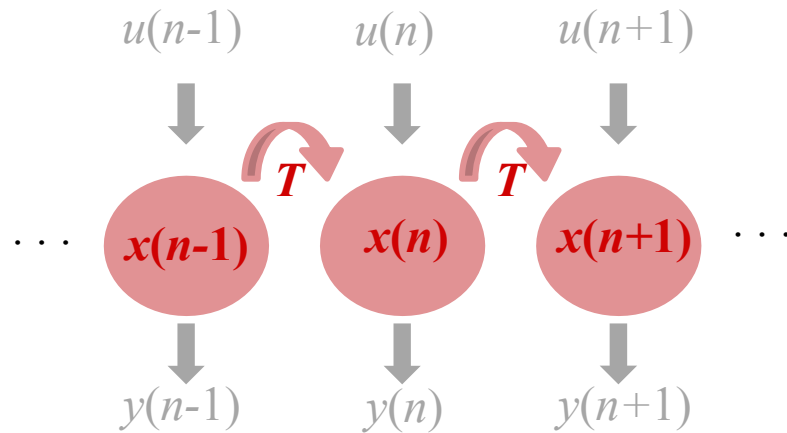
- physicists / neurophysiologists / neurologists / neurolinguists want to model neural architectures / mechanisms

- tools from dynamical systems theory proper, "ABC" tradition, stochastic processes and information theory

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

$\cdots$ $\cdots$

$y(n\text{-}1)$ $y(n)$ $y(n\text{+}1)$
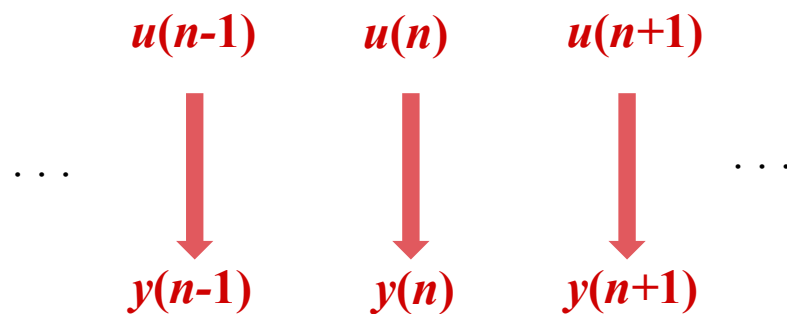
- cognitive scientists / behavioural neuroscientists / linguists / BCI engineers want to model neural function / performance

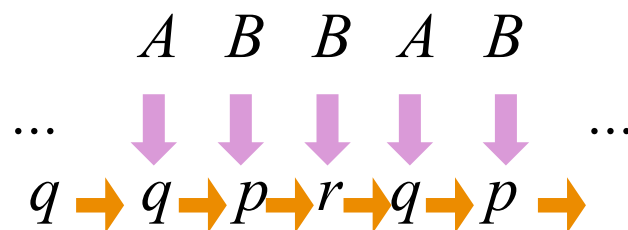- tools from signal processing, machine learning, CS, stochastic processes and information theory

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

Part 2: A Zoo of Finite-State Models

JACOBS
UNIVERSITY

Deterministic finite-state automata (DFA)
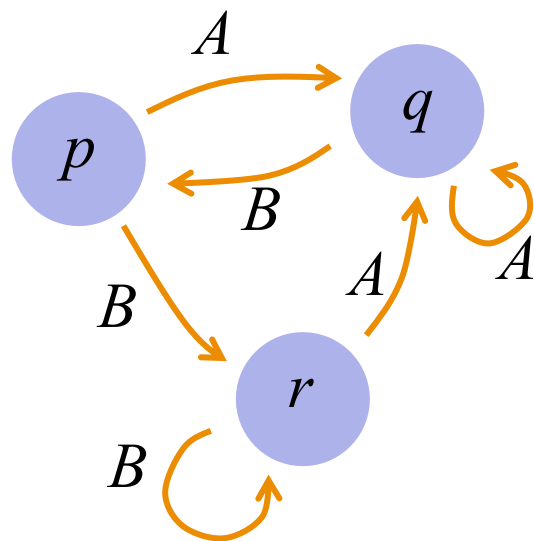
example trajectory:

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

... ⬇ ⬇ ⬇ ⬇ ⬇ ...

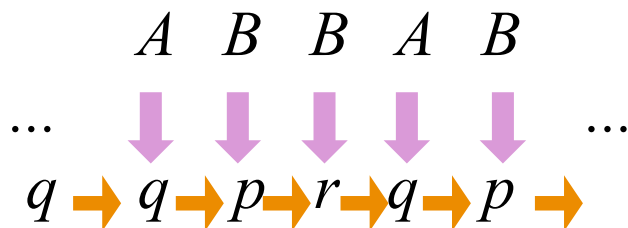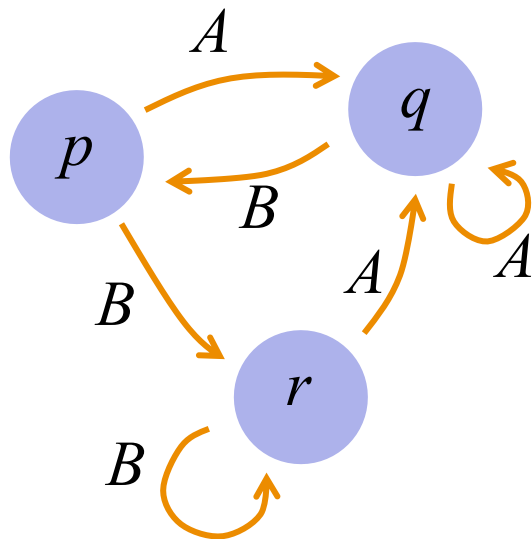$q \rightarrow q \rightarrow p \rightarrow r \rightarrow q \rightarrow p \rightarrow$

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 $T: Q \times \Sigma \rightarrow Q$, 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")

JACOBS
UNIVERSITY

Moore- and Mealy-Machines
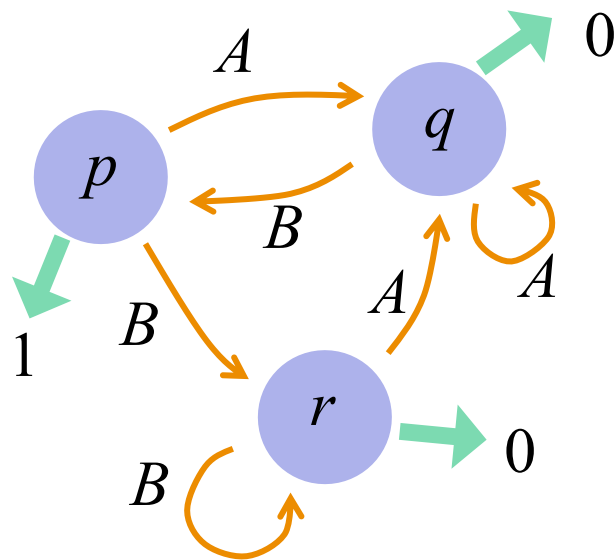
example Moore trajectory:
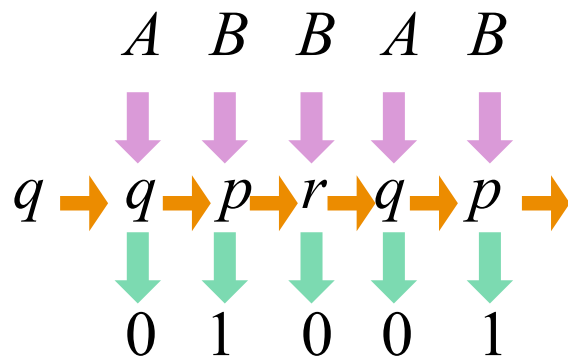
A Moore Machine is a DFA, equipped additionally with

- a finite set of output symbols, e.g. $O = \{0, 1\}$
- a translation (observation) function $\rho: Q \to O$

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

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

State-of-the-art entrance paper for learning Mealy machines from data:

Steffen, B., Howar, F., & Merten, M. (2011). Introduction to active automata learning from a practical perspective. In *Formal Methods for Eternal Networked Software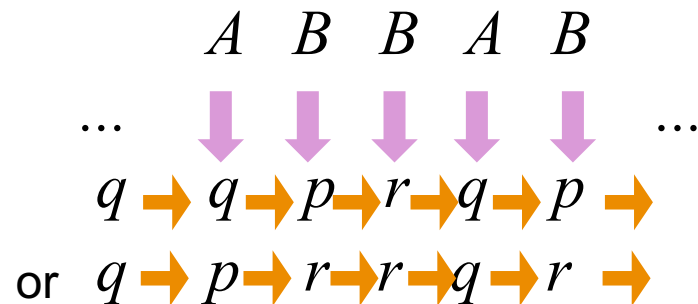 Systems* (pp. 256-296). Springer Berlin Heidelberg. http://ls5-www.cs.tu-dortmund.de/cms/en/research/papers/introduction-to-automata-learning-sfm2011.pdf

JACOBS
UNIVERSITY

Non-deterministic finite-state automata (NFA)
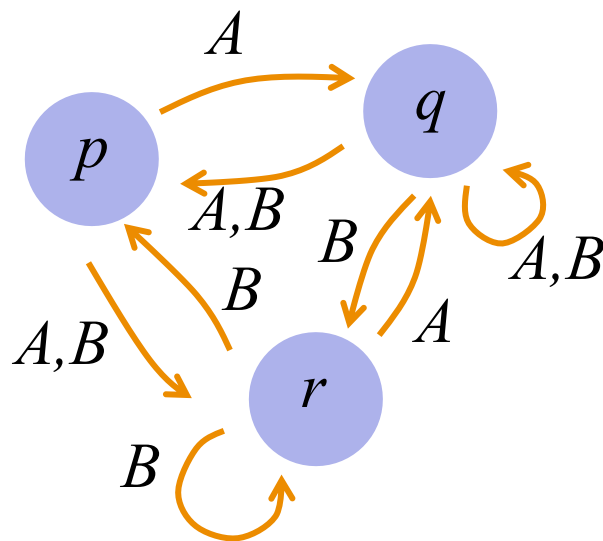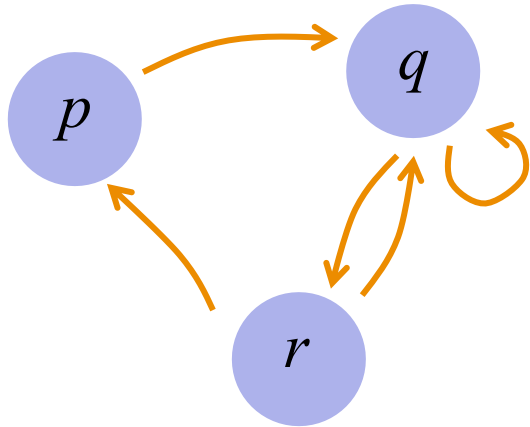
A NFA is defined by:

- 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 $T\colon Q \times \Sigma \to \mathrm{Pot}(Q)$, (Pot: power set), e.g.

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

example trajectories:

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

$$\ldots \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \ldots$$

$$q \to q \to p \to r \to q \to p \to$$

or $\quad q \to p \to r \to r \to q \to r \to$

JACOBS
UNIVERSITY

NFAs, comments

special case: no input symbols

example trajectory:

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

- 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)

JACOBS UNIVERSITY

Finite-dimensional Markov chains
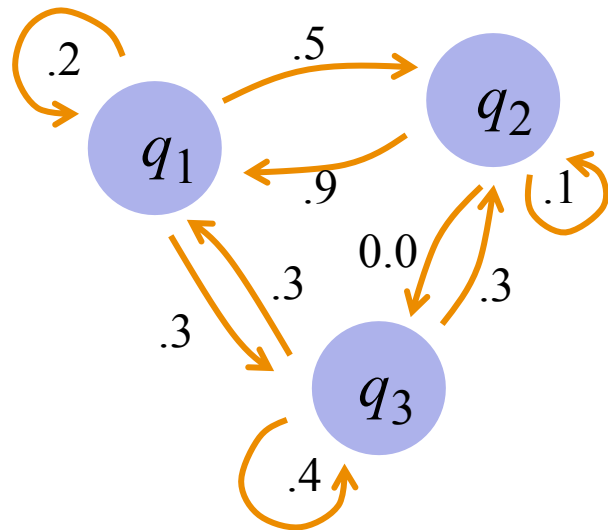

A (finite-dimensional) Markov chain (MC) is defined by:

- 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

JACOBS
UNIVERSITY

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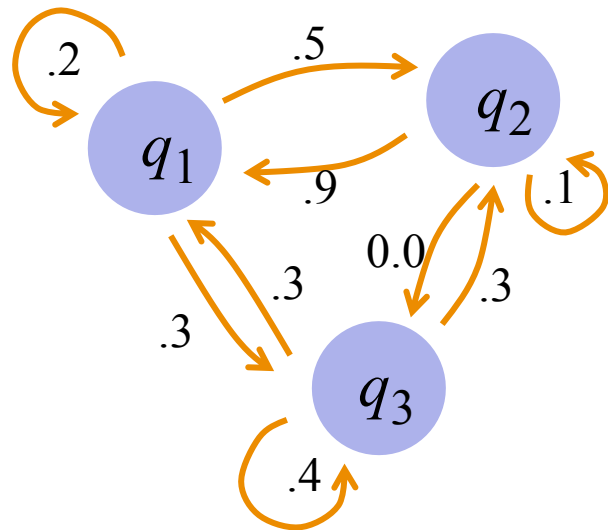
$$\boldsymbol{p} = \begin{pmatrix} 0.5 \\ 0.0 \\ 0.5 \end{pmatrix}$$

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

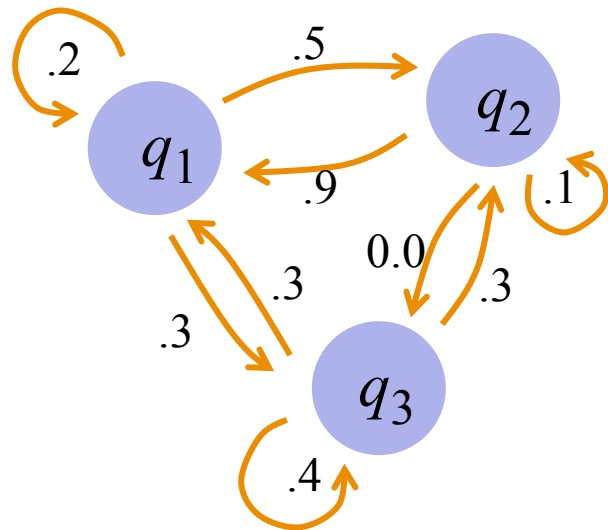- probability that a trajectory starts with

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

is

$$P(X_0 = q_{i_0}, ..., X_n = q_{i_n}) =$$
$$= P(X_0 = q_{i_0}) \cdot P(X_1 = q_{i_1} \mid X_0 = q_{i_0}) \cdot ... \cdot$$
$$\cdot P(X_n = q_{i_n} \mid X_{n-1} = q_{i_{n-1}})$$

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

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

- 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 $q_{i_{n+1}}$) only depends on current state q_{i_n}, not on previous state history

- MCs are "memoryless" systems

JACOBS
UNIVERSITY

Controlled Markov chains
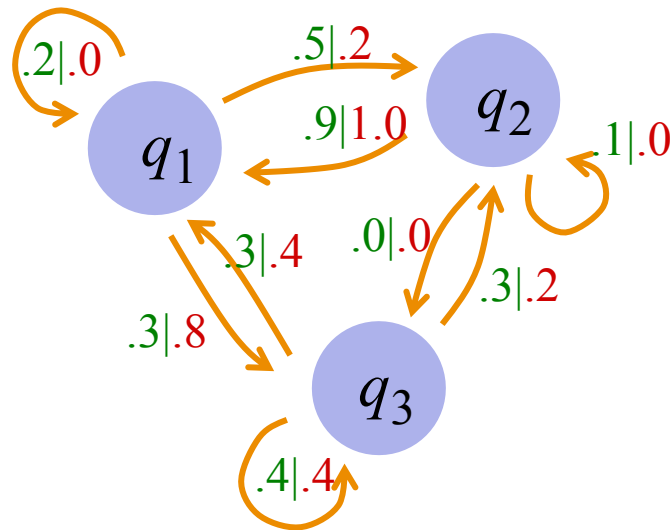
$$p = \begin{pmatrix} 0.5 \\ 0.0 \\ 0.5 \end{pmatrix}$$

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. $A = \{a_1, a_2\}$

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

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

T_{a_1}	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_{a_2}	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

JACOBS
UNIVERSITY

Hidden Markov models (HMMs)
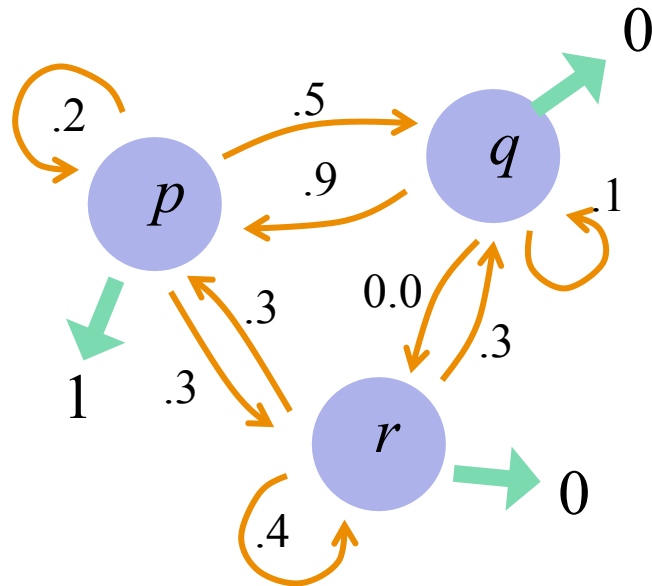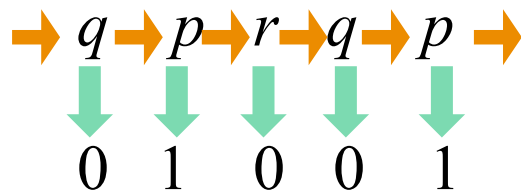
Defining components:

- 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
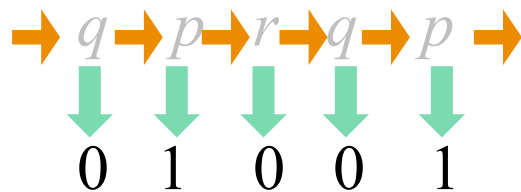
    or, equivalently,

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

example trajectory:

JACOBS
UNIVERSITY

HMMs, comments 1
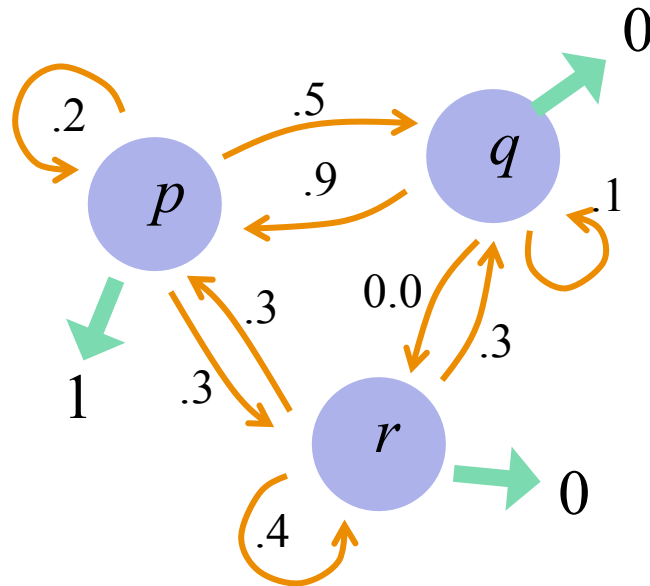
unobservable states

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

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"

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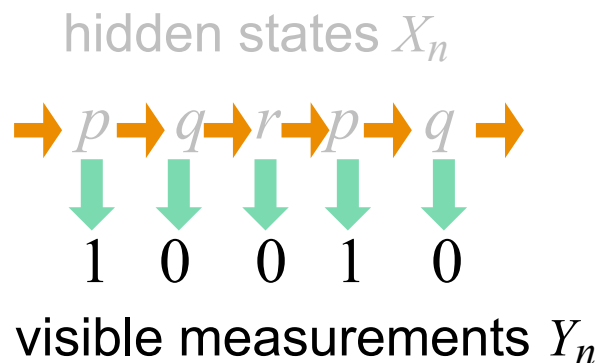
HMMs, comments 2
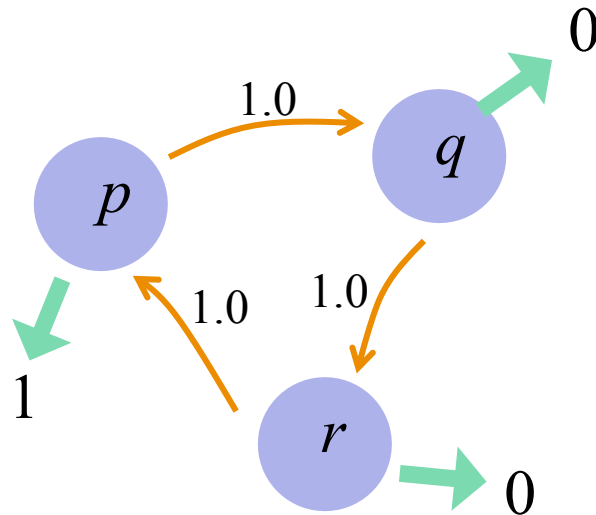
- The visible trajectories (values of random variables Y_n) "have memory":

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

- 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

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

1   0   0   1   0

visible measurements Y_n

Literature

The classical tutorial text on HMMs, very readable (20000+ 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)

JACOBS
UNIVERSITY

Controlled hidden Markov models, aka POMDPs
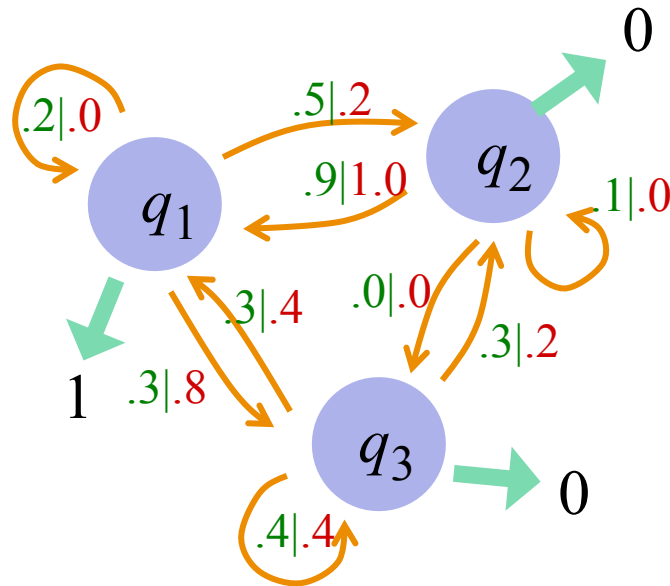
- crossover of controlled MCs and HMMs
- also (widely) known as Partially Observable Markov Decision Processes (POMDPs)
- a basic tool in theory of autonomous agents / robotics / reinforcement learning in the machine learning sense
- In that context, a POMDP constitutes the agent's world model:
  - Q: 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
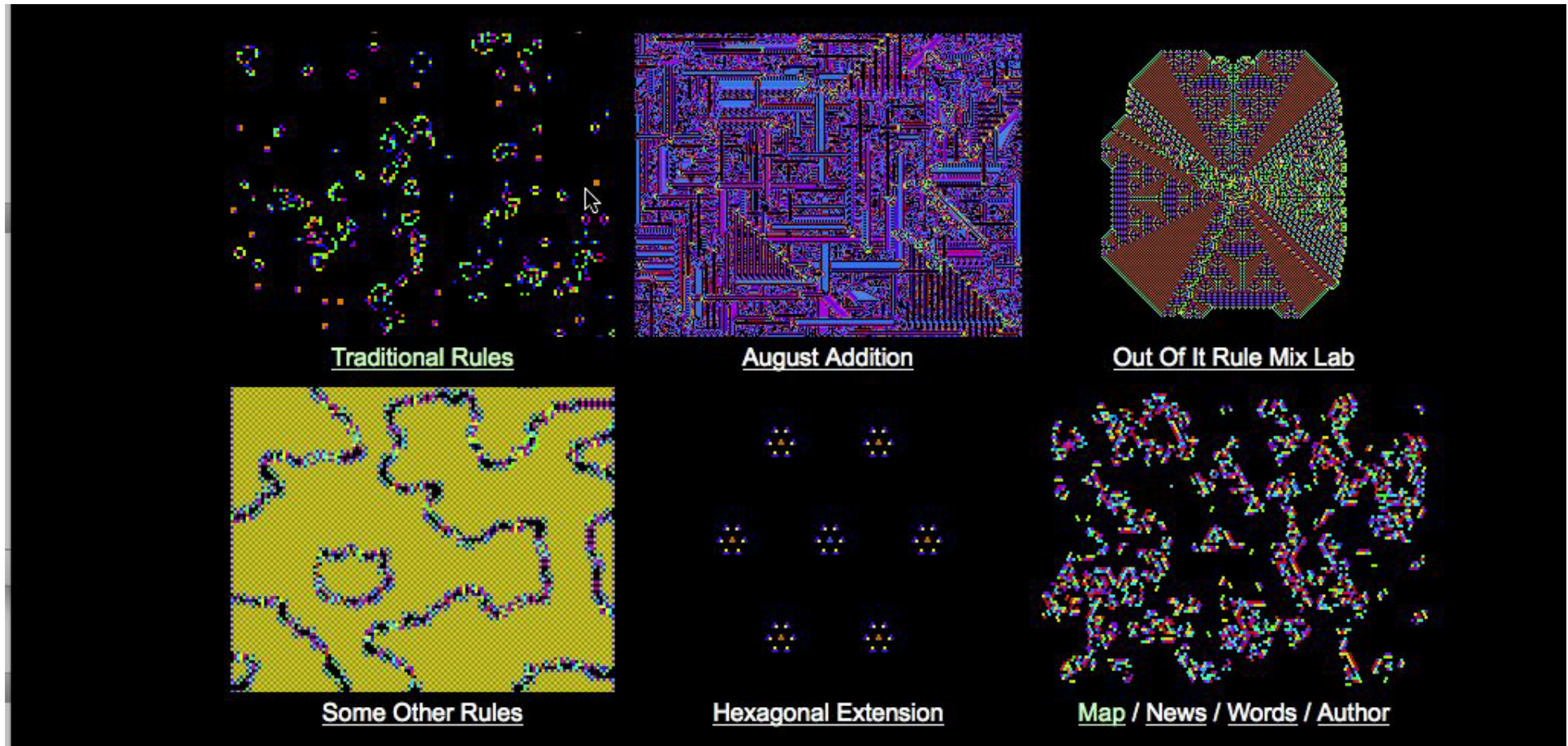
IK 2018

Dynamical Systems: a Navigation Guide

Herbert Jaeger

JACOBS
UNIVERSITY

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)

JACOBS UNIVERSITY

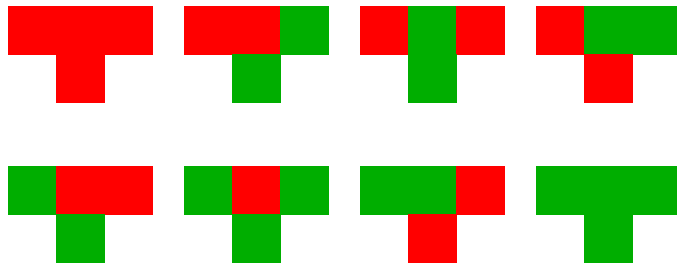# 1-dim deterministic CA

Defining components:

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

- a local transition function

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

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 $T: \{\text{configurations}\} \to \{\text{configurations}\}$

JACOBS UNIVERSITY

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 τ: $Q \times Q \times Q \rightarrow Prob(Q)$
- 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
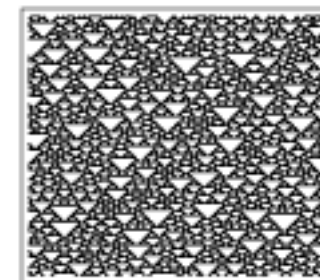 - computational power analysis of neural circuits

Fowler et al, 1992

Wolfram Class 1   Class 2

Class 3   Class 4

Wolfram, 2002

JACOBS
UNIVERSITY

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

JACOBS
UNIVERSITY

Petri nets

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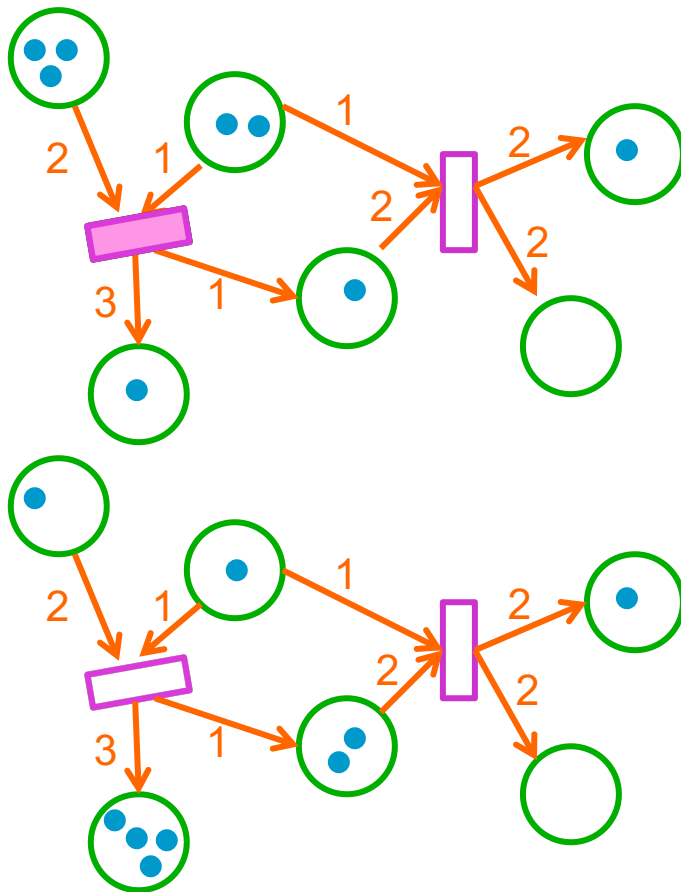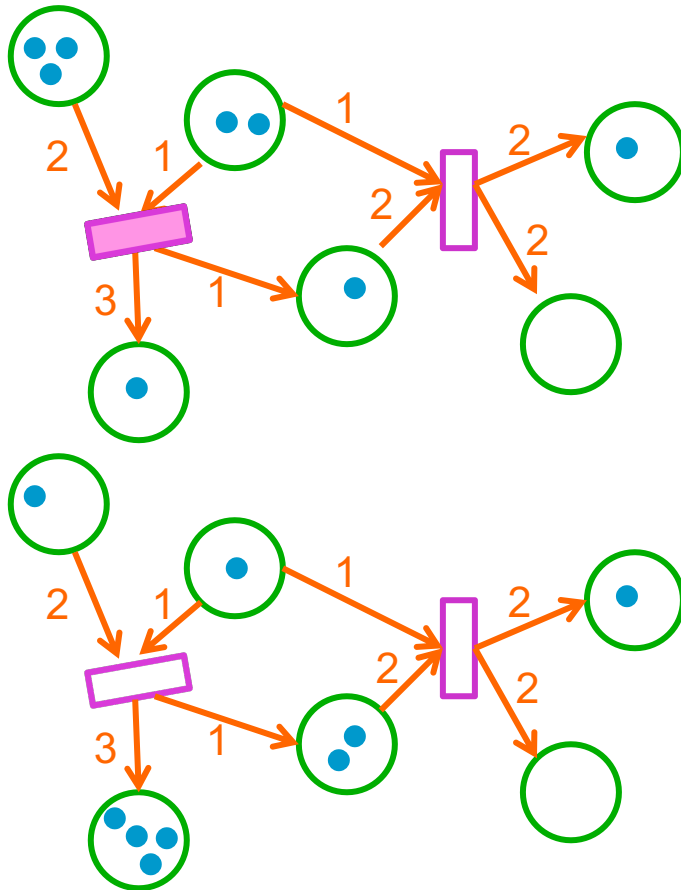

Components:

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?

JACOBS UNIVERSITY

Literature

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

# Dynamical Bayesian networks, dynamical graphical models
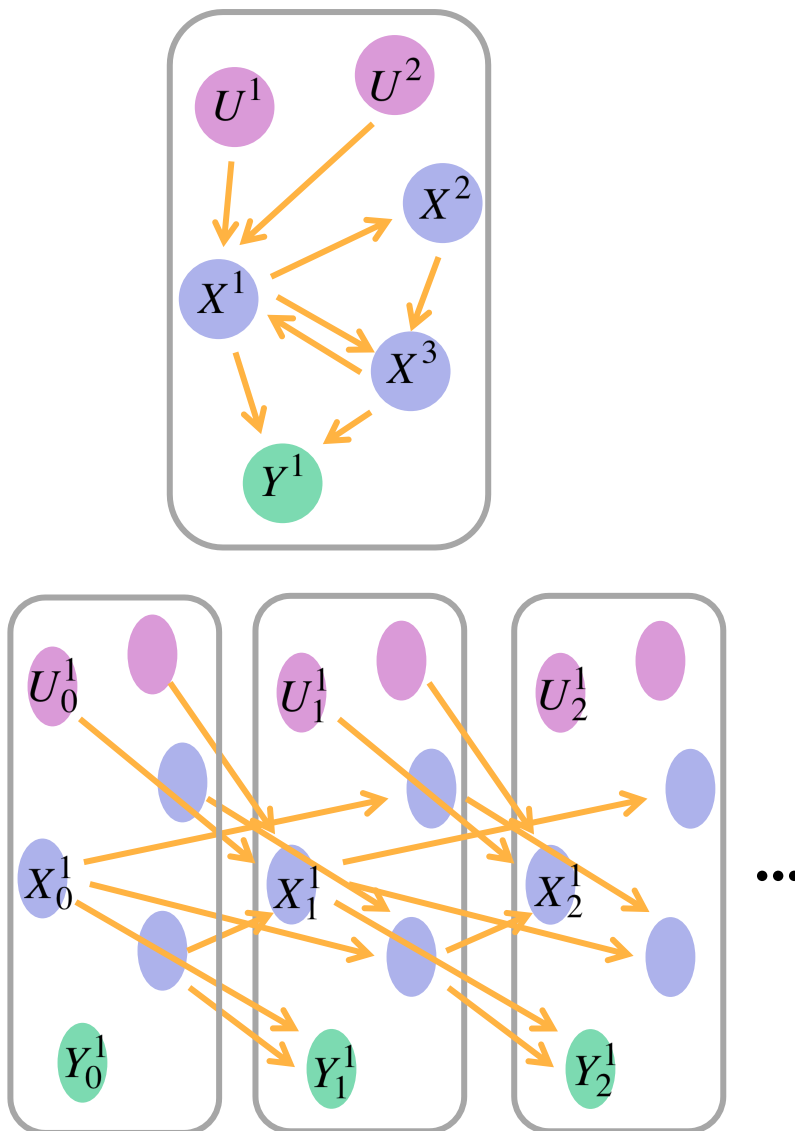
- Idea: describe the state of a stochastic DS at time n by the values of a finite collection of finite-valued
  - input random variables (RVs) U_n^i
  - hidden RVs X_n^j
  - output RVs Y_n^k

- 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

JACOBS
UNIVERSITY

Dynamical Bayesian networks, comments
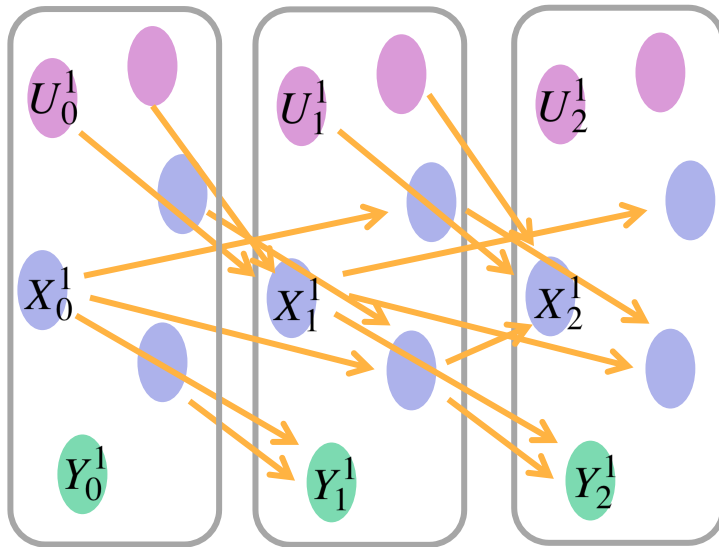
- DBNs can be seen as multivariate generalization of HMMs and POMDPs

- 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

JACOBS UNIVERSITY

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 current 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

JACOBS UNIVERSITY

Part 3: A Ménagerie of Continuous-State Models

JACOBS
UNIVERSITY

Iterated function systems, aka iterated maps

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

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

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

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

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

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

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

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

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

- discrete time steps $n \in \mathbb{Z}$ or $n \in \mathbb{N}$

- update operator is a function

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

We will inspect such systems in more detail later

JACOBS
UNIVERSITY

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.

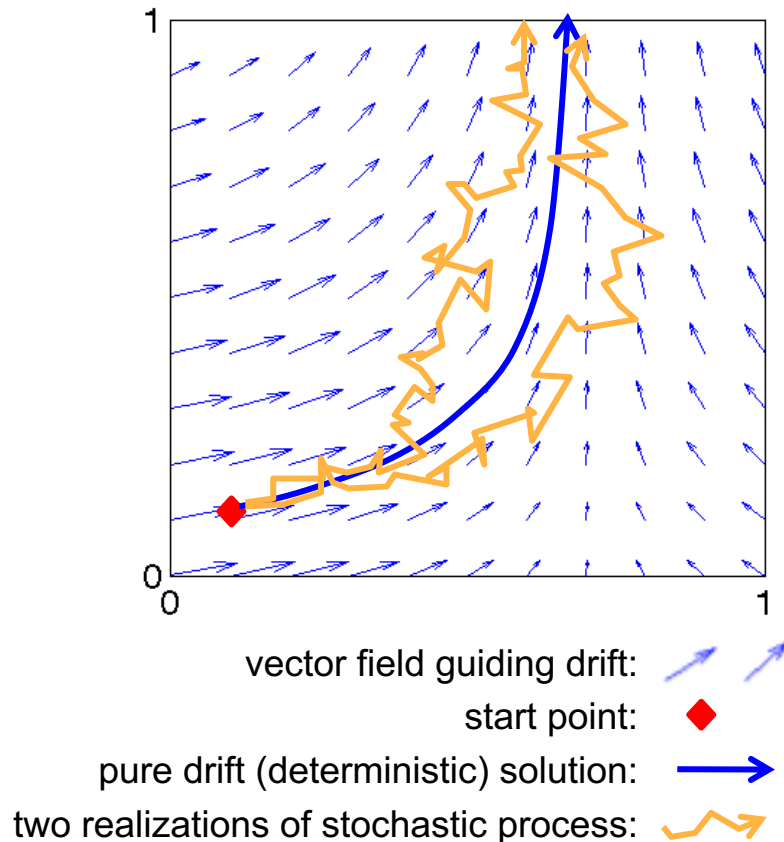$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t)$$

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

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

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

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

- update operator is a function
  $$T \colon \mathbb{R}^N \to \mathbb{R}^N$$

- 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

JACOBS
UNIVERSITY

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)

JACOBS
UNIVERSITY

Literature

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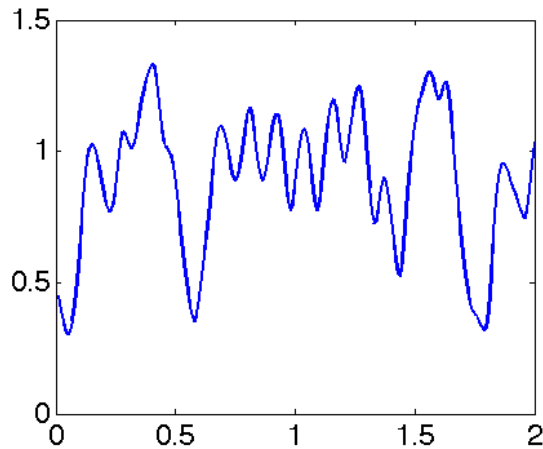points to concepts and literature.

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

JACOBS
UNIVERSITY

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

JACOBS
UNIVERSITY

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

JACOBS
UNIVERSITY

Partial differential equations (PDEs)

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

JACOBS
UNIVERSITY

(Neural) field equations (FEs)
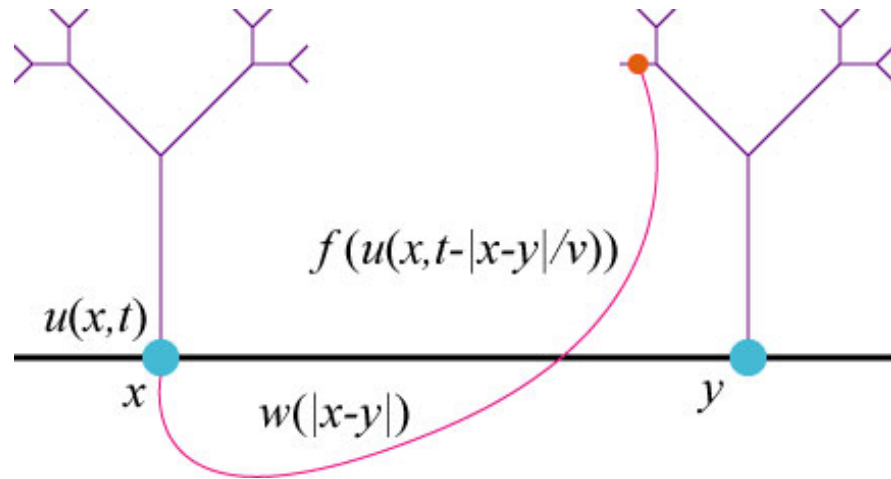
$$c \frac{\partial u(x,t)}{\partial t} = -u + \int\limits_{-\infty}^{+\infty} dy \, w(|x-y|) \, f(u(x-y, t - |y|/v))$$

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

- 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

JACOBS UNIVERSITY

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

JACOBS
UNIVERSITY

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

JACOBS
UNIVERSITY

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

JACOBS
UNIVERSITY

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?

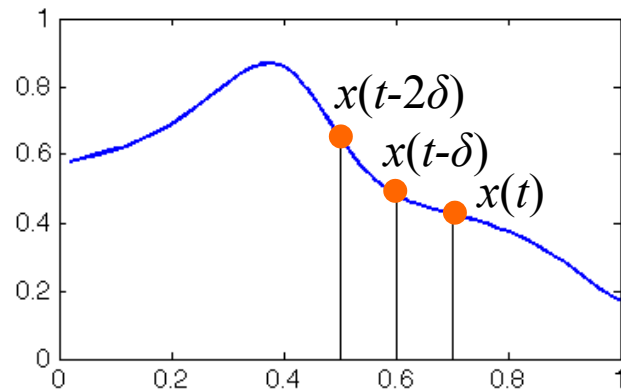That's food for thought for people like you and me.

JACOBS
UNIVERSITY

Creating state representations by delay embeddings

given: 1-dim timeseries $x(t)$

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

turn into 3-dim timeseries
$\mathbf{x}(t) = (x(t), x(t\text{-}\delta), x(t\text{-}2\delta))'$
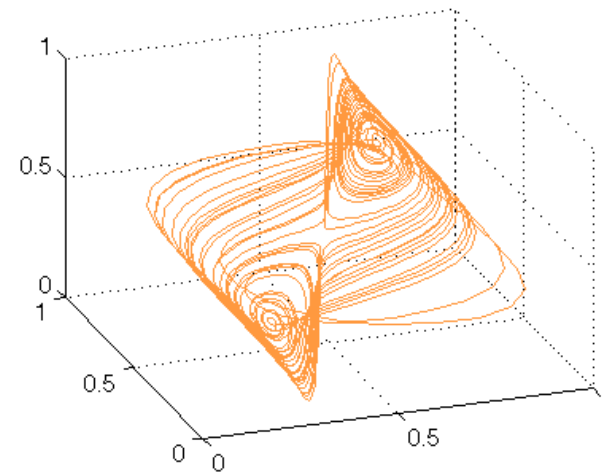
- we can artificially transforme a 1-dim state into an m-dim state by time-delay embedding

- no information added

- what is the "real" system state?

JACOBS
UNIVERSITY

Takens' theorem (simplified)

Preparation: A d-dimensional manifold is a subset of $\mathbb{R}^n$ that locally is d-dimensional. Example:

- unit sphere surface: 2-dim manifold in $\mathbb{R}^3$

Theorem of Takens

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

- 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

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

Then, $\mathbf{e}(t)$ evolves on a d-dimensional manifold $M' \subseteq \mathbb{R}^{2d+1}$, 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

$\mathbf{e}(t)$

JACOBS UNIVERSITY
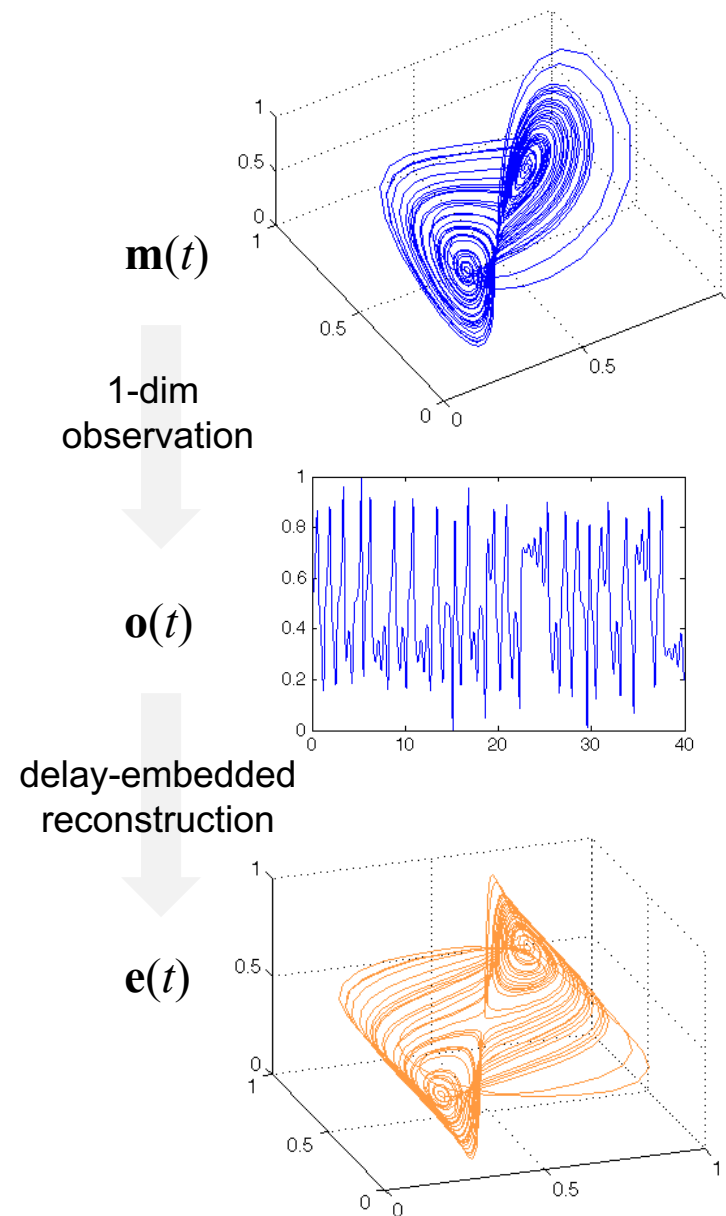
Takens' theorem, explanations

- "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?

$\mathbf{m}(t)$

$\mathbf{o}(t)$

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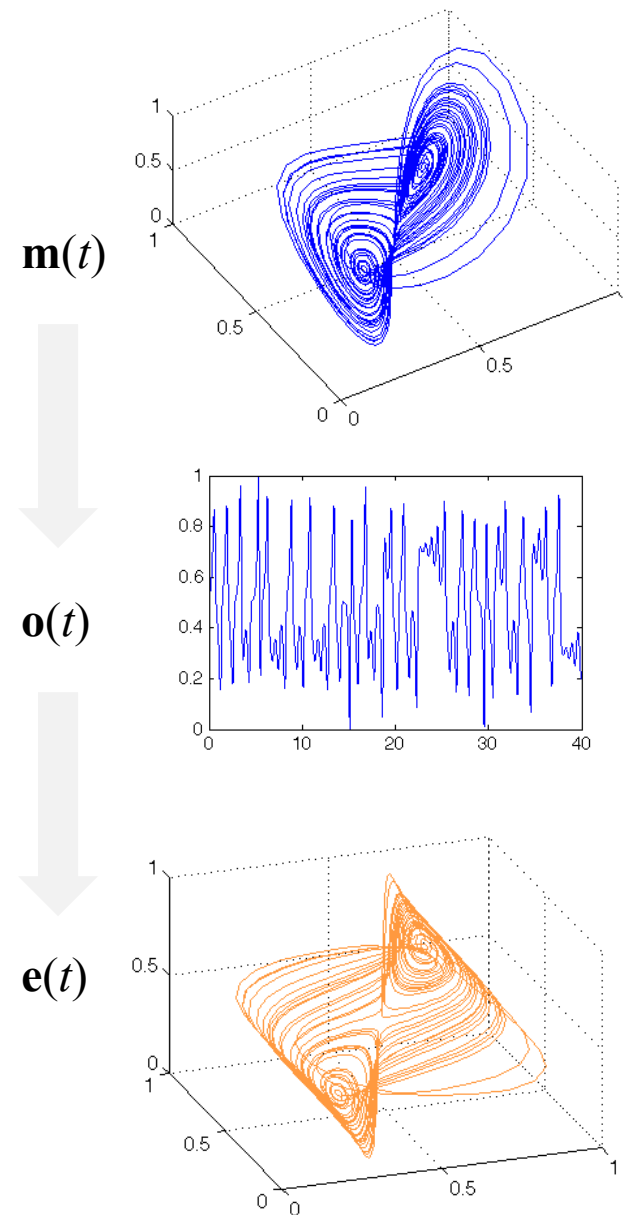
Takens' theorem, comments

- 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.

$\mathbf{m}(t)$

$\mathbf{o}(t)$

$\mathbf{e}(t)$

JACOBS UNIVERSITY
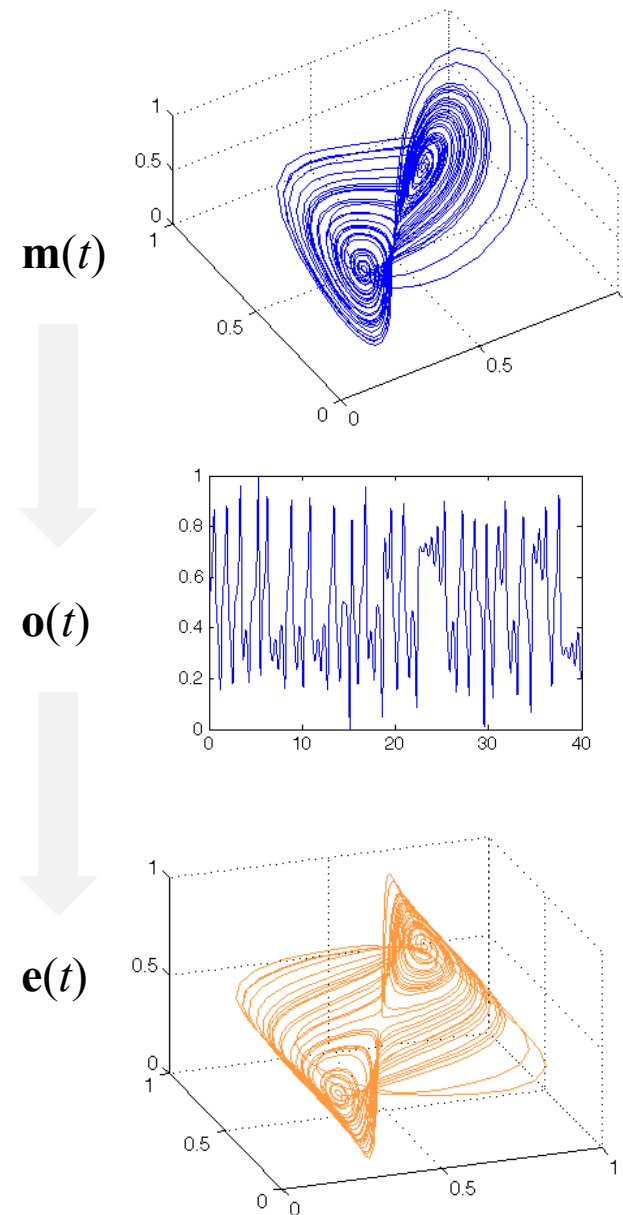
Literature

The original paper of Takens (>8500 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

JACOBS
UNIVERSITY

IK 2018

Dynamical Systems: a Navigation Guide

Herbert Jaeger

JACOBS
UNIVERSITY

Back to discussing states...

state $\mathbf{x}(0)$

observations $o(n)$

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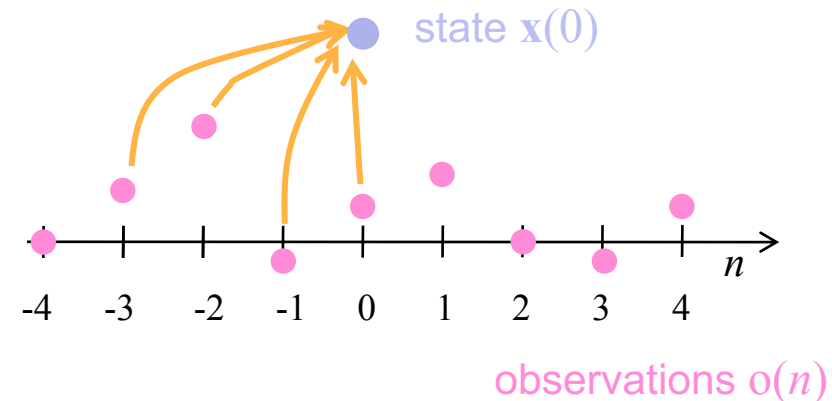
$$P(X_{n+1} = a \mid X_0 = b_0, ..., X_n = b_n)$$

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

JACOBS
UNIVERSITY
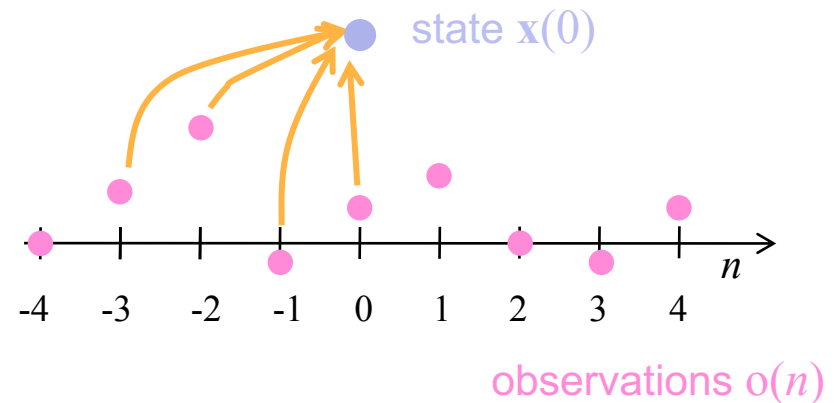
Turning the tide of time

- 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
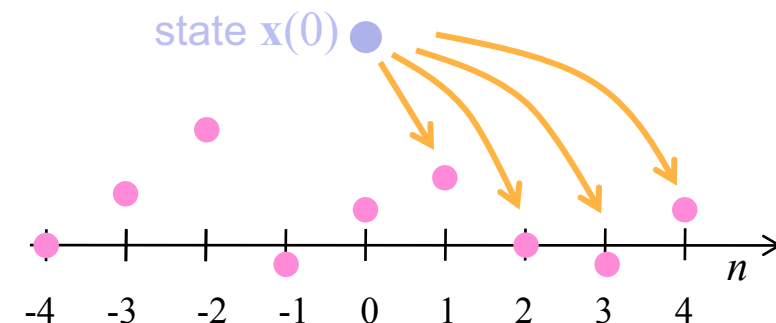
" ... a state of a system at any given time is the information needed to determine the behaviour of the system from that time on." L. A. Zadeh, 1969

JACOBS
UNIVERSITY

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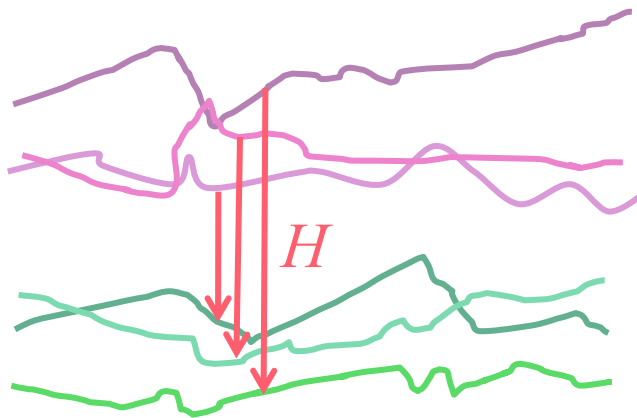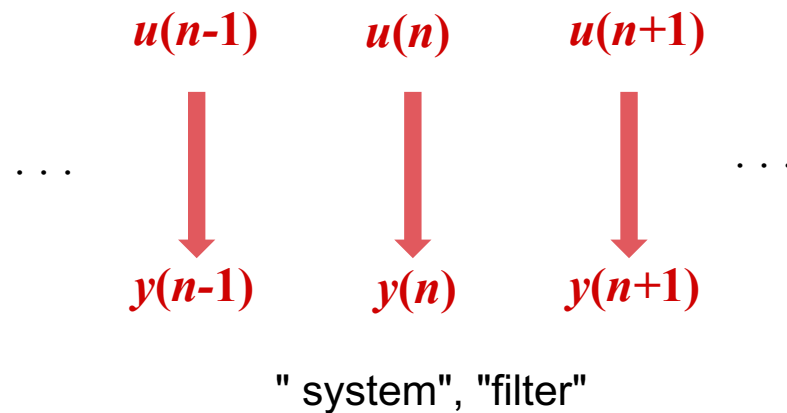
L.A. Zadeh. The concept of system, aggregate, and state in system theory. In L.A. Zadeh and E. Polak, editors, System Theory vol 8, 3 - 42. McGraw-Hill, New York, 1969.

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

JACOBS
UNIVERSITY

Part 5: State-Free Modeling of Temporal Systems

JACOBS
UNIVERSITY

The classical engineering view on "systems"

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

$$\cdots \qquad\qquad\qquad\qquad\qquad \cdots$$

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

" 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:

  $s\colon \mathbb{Z} \longrightarrow \mathbb{R}$ (or $\mathbb{C}$)   *discrete time* or

  $s\colon \mathbb{R} \longrightarrow \mathbb{R}$ (or $\mathbb{C}$)   *continuous time*

  (suitably constrained, e.g. finite variance)

- Signals are just "time series"

- Let $\mathcal{S}$ be the set of signals (discrete or continuous time, suitably constrained)

- A filter (or system) is a functional

  $H\colon \mathcal{S} \longrightarrow \mathcal{S}$

- 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
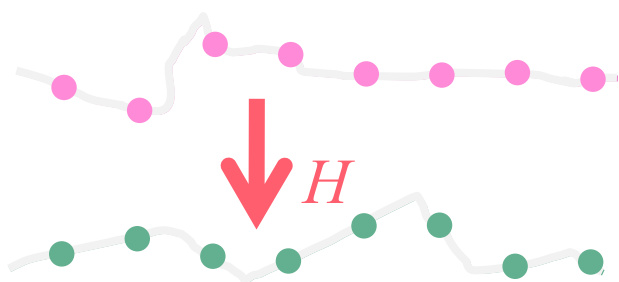
- The historically earlier theory concerned continuous-time analog signals and filters.

- 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.

JACOBS
UNIVERSITY

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.

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

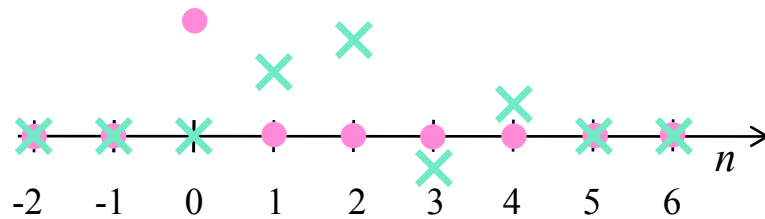- A system is **causal** if current output depends only on previous history

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

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

LTI systems

$$\delta(n) = \begin{cases} 1, & \text{if } n = 0 \\ 0, & \text{else} \end{cases}$$
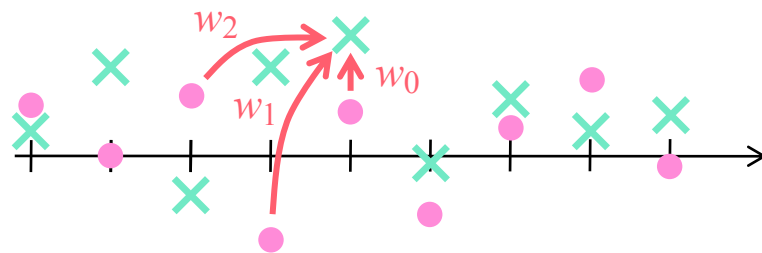
- The unit impulse δ is the signal that is zero everywhere except at $n = 0$, where it is 1

- 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 $s(t)$:

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

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)

JACOBS
UNIVERSITY

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

Advanced undergraduate-level lecture notes, more comprehensive than the previous: Richard Baraniuk, Signals and Systems – lecture notes. Rice University.
http://www.eng.ucy.ac.cy/cpitris/courses/ece623/notes/SignalsAndSystems.pdf

JACOBS
UNIVERSITY

Literature, continued

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

JACOBS
UNIVERSITY

Context-free grammars

Example

$V = \{S, A\}$

$T = \{(,)\}$

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

$\quad A \to AA$

A derivation:

S
$\downarrow$
A
$\downarrow$
A

$A \qquad A$

$\qquad A$

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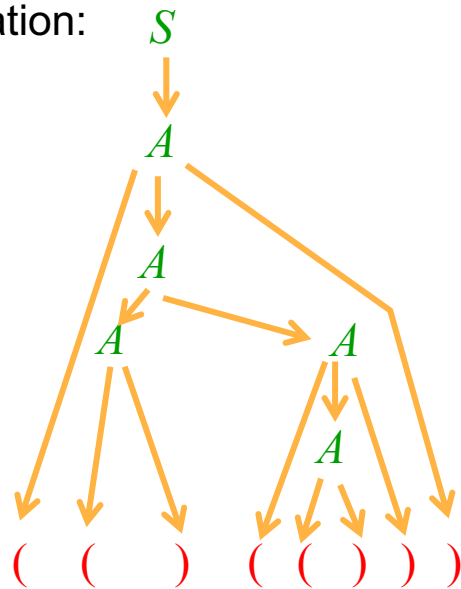
Due to this derivation, the word $(()(()))$ is in this grammar's language.

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

JACOBS
UNIVERSITY

Grammars, automata, Chomsky hierarchy

The Chomsky hierarchy

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

  $\subset\!\!\!\!\subset$

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

  $\subset\!\!\!\!\subset$

  . . .

  $\subset\!\!\!\!\subset$

- 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

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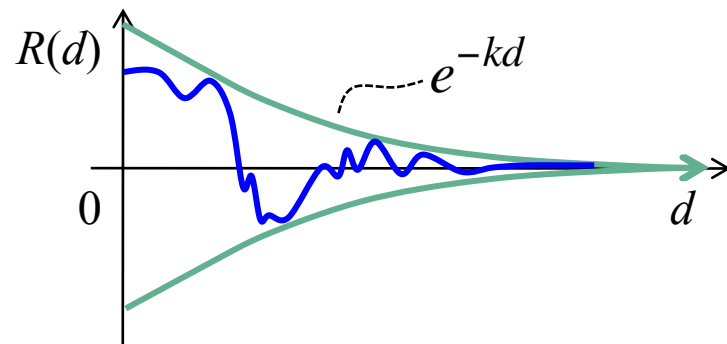
- Consider the autocovariance $R(d)$ of a numerical timeseries $x(t)$ generated by a stationary process

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

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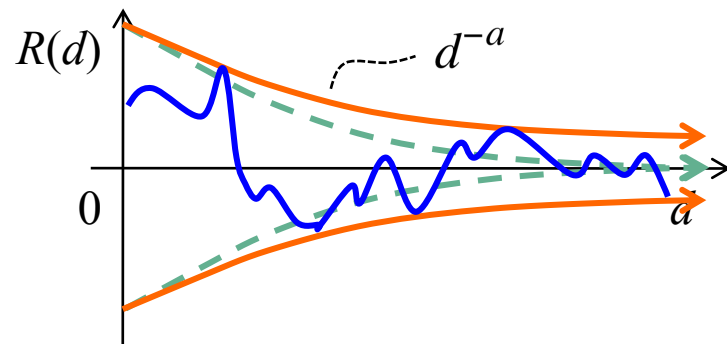
$$\dot{x}(t) = T(x(t), u(t)) \quad \text{or}$$

$$x(n + 1) = T(x(n), u(n + 1))$$

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

JACOBS
UNIVERSITY

"Long-range" interaction: power laws

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

- 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
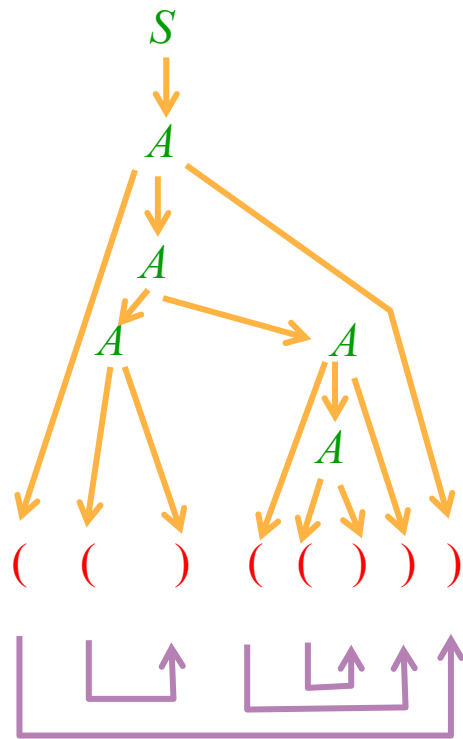
Nested dependencies, long-range effects at different timescales, no single characteristic timescale

- 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://svn.eecs.jacobs-university.de/svn/eecs/archive/bsc-2011/jdjolonga.pdf

JACOBS
UNIVERSITY

Literature, continued

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

JACOBS
UNIVERSITY

Part 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.

JACOBS
UNIVERSITY

Our tutorial example

System equation of a 2-dimensional DS in polar coordinates r and θ :

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

$$\dot{\theta} = 1$$

- Describes a motion of a point (given by coordinates r and θ) in $\mathbb{R}^2$
- r and θ are the state variables. The set of all possible values of the state variables (here: $\mathbb{R}^2$) is the state space (also: phase space)
- a is a (control) parameter, can be set by the "user"; each setting yields a different version of the system equation

JACOBS
UNIVERSITY

Vector fields

$$\dot{r} = -r^3 + r \quad \text{[we use } a = 1 \text{ for a while]}$$

$$\dot{\theta} = 1$$

- Use Cartesian coordinates x, y for plotting

- $x = r \cos \theta$, $y = r \sin \theta$

- At each point (x, y), consider the vector $(\dot{x}, \dot{y})$

- Gives a vector field

- Each vector describes a velocity + direction of a local motion (velocity field)

y
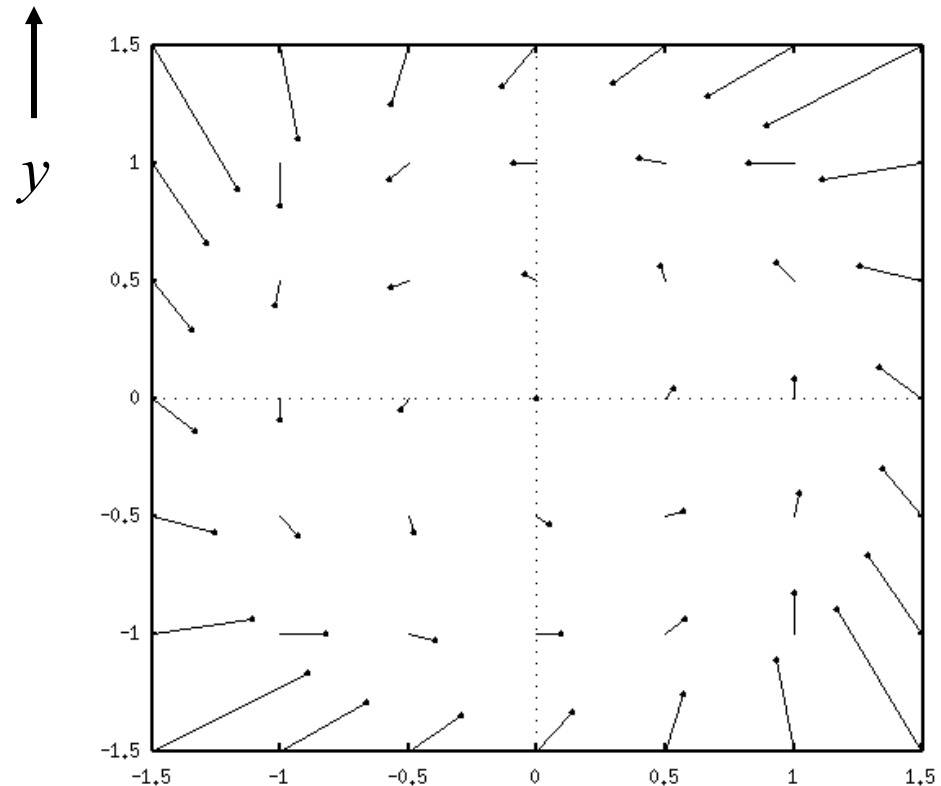
Note: in the plot, velocity vectors are scaled down to prevent clutter

x

For all plots in this section I used Bard Ermentrout's tool XPPAUT5.41

JACOBS
UNIVERSITY

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 (http://www.math.pitt.edu/~bard/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 (difficult!).

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.

JACOBS
UNIVERSITY

A closeup look

State ⊙ :

$(x, y) = (1,1)$

$(r, \theta) = (\sqrt{2}, \pi/4$

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

$\dot{r}[(1,1)] = -\sqrt{2}^3 + \sqrt{2}$

$= -\sqrt{2}$

$\dot{\theta} = 1 \Rightarrow$

$(\dot{\theta}r)[(1,1)] = \sqrt{2}$
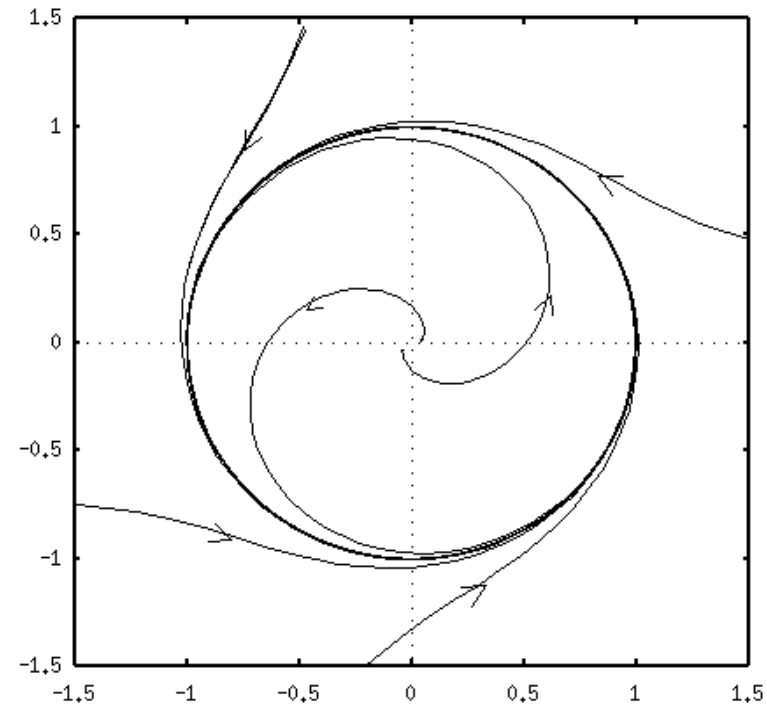
$r = \sqrt{2}$

$\theta = \pi/4$

JACOBS
UNIVERSITY
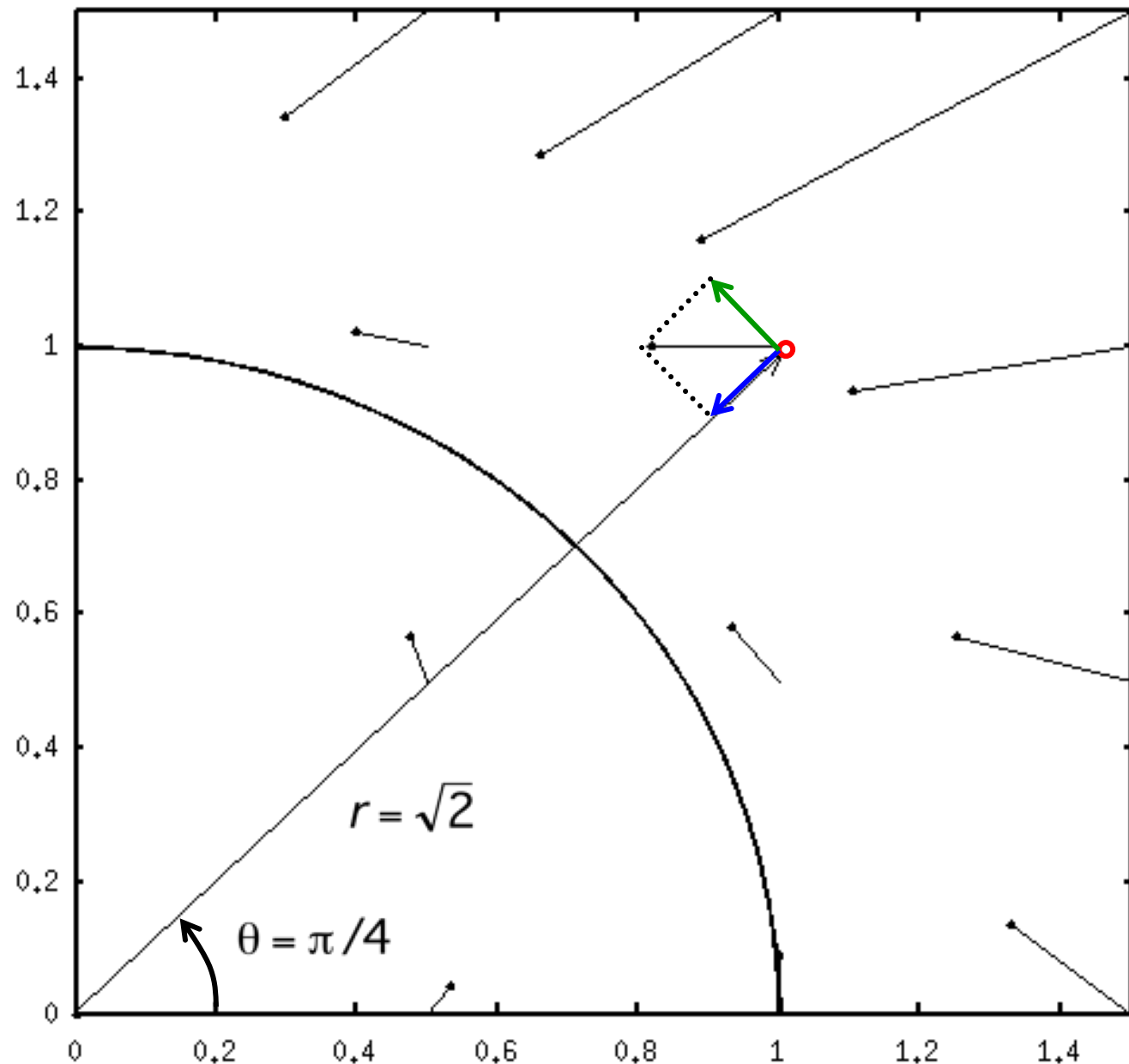
Trajectories and phase portraits

$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$
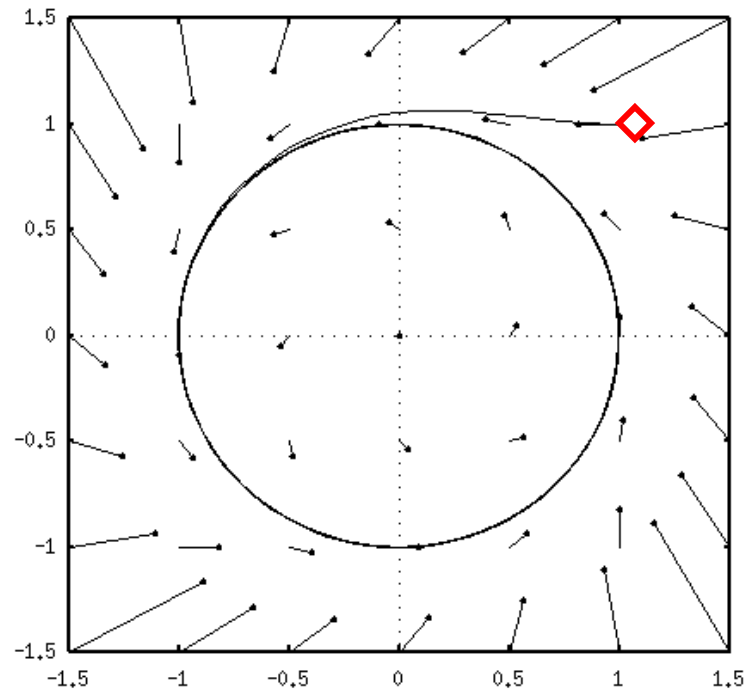
Following the vector field from any starting point ◇ gives a trajectory.

Plotting several characteristic trajectories gives a phase portrait.

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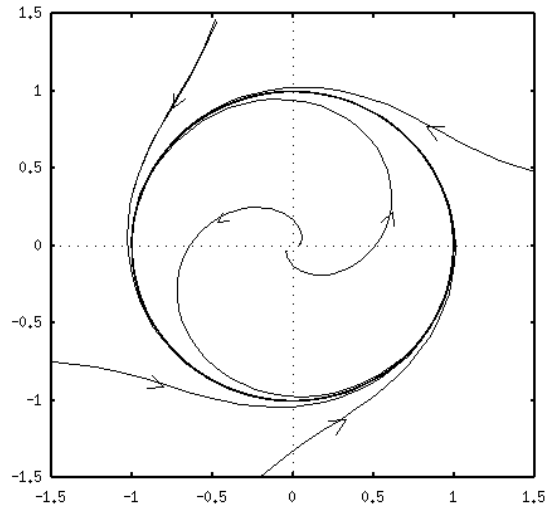
$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$

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 $\dot{r} = -r^3 + r$
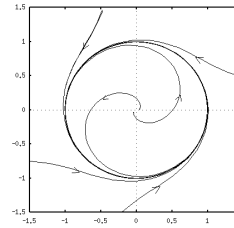
$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$
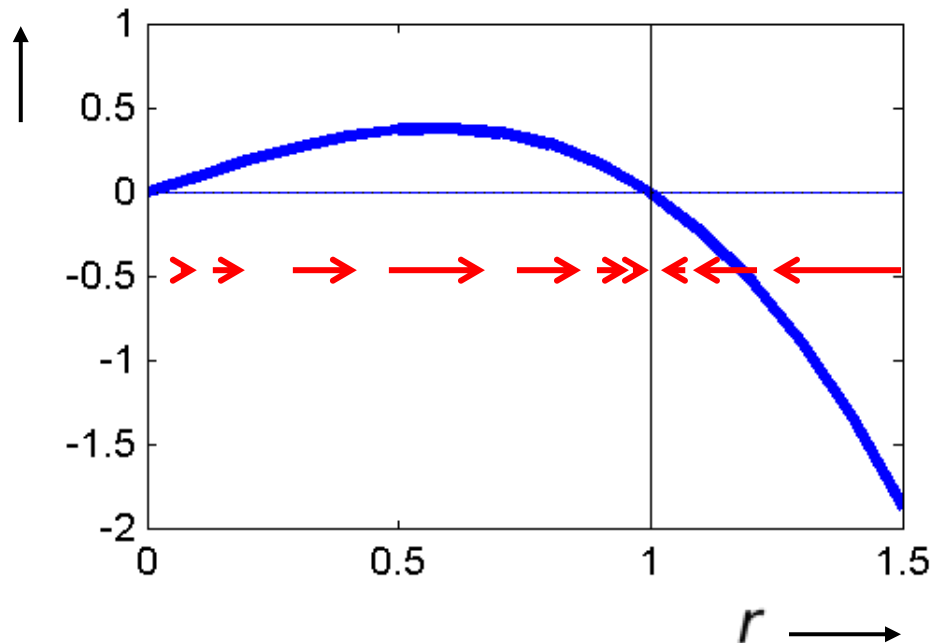
- For $0 < r < 1$, we have $\dot{r} > 0$, so r will grow with time.

- 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.

JACOBS UNIVERSITY

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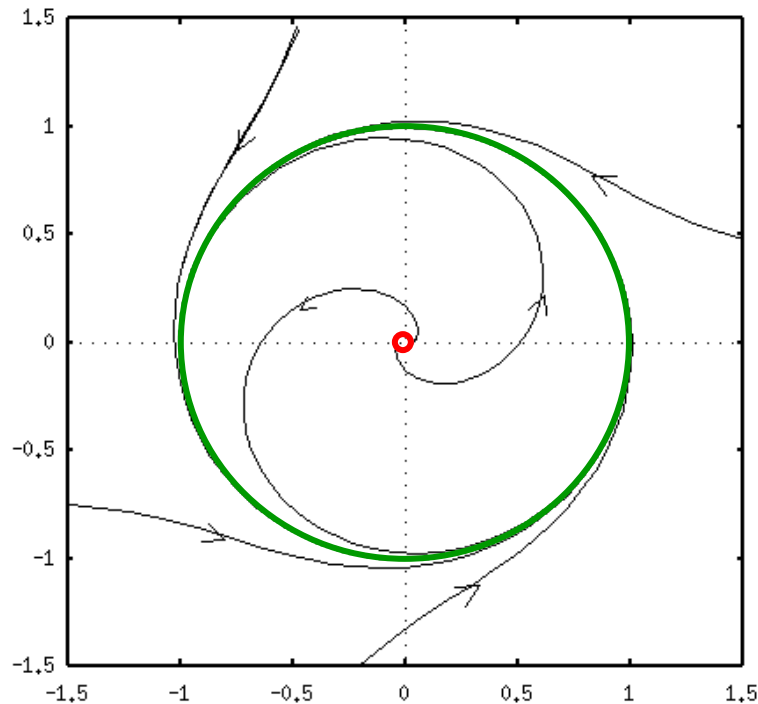
$$\dot{r} = -r^3 + r; \quad \dot{\theta} = 1$$

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.
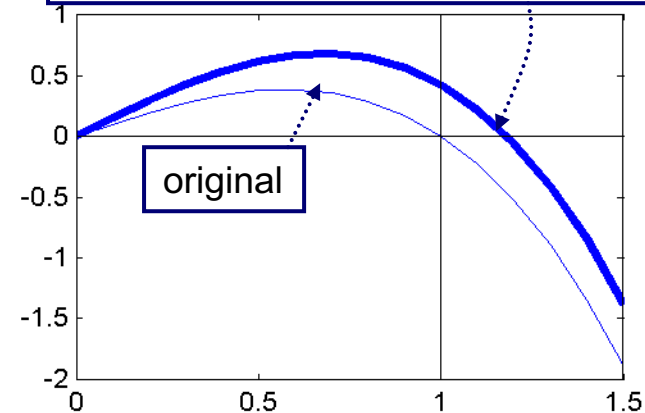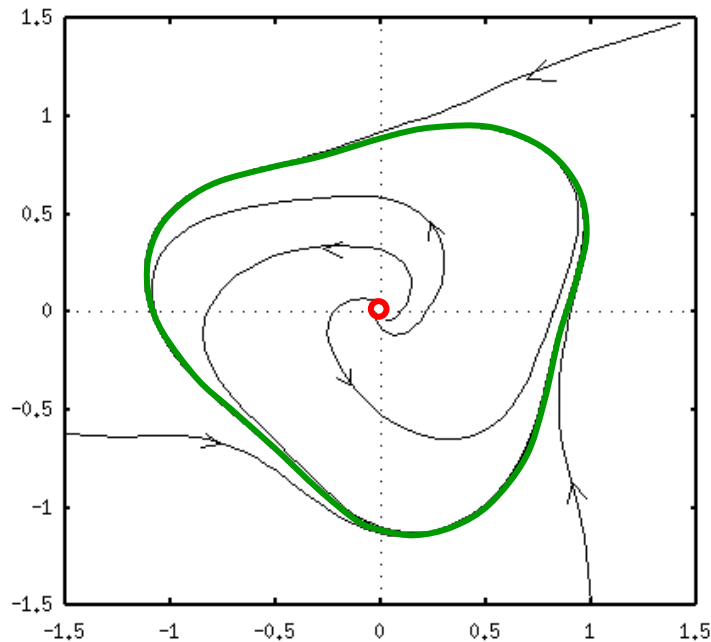
Surprise!
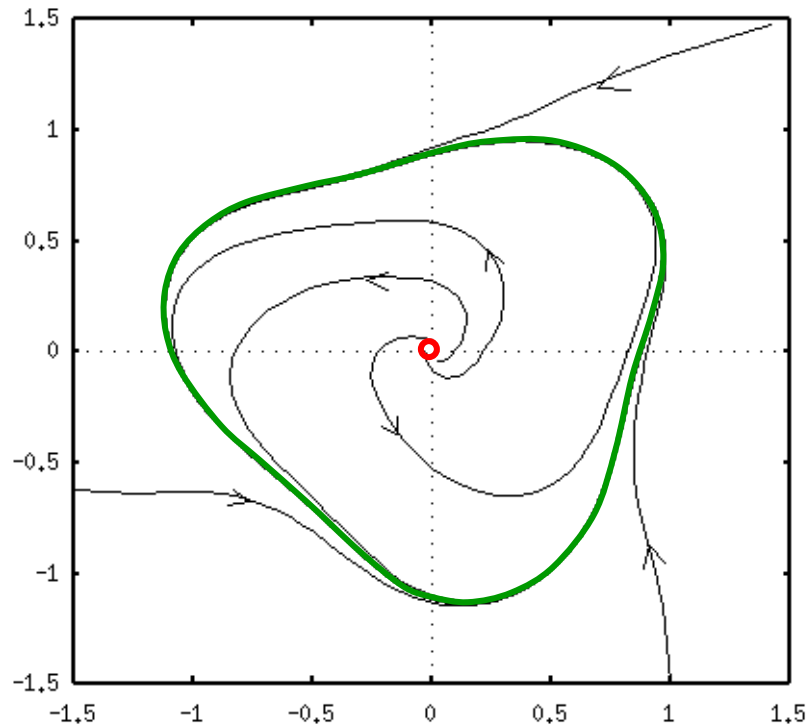
We added some extra terms to the RHS of the system equations.

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

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

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

JACOBS
UNIVERSITY

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.
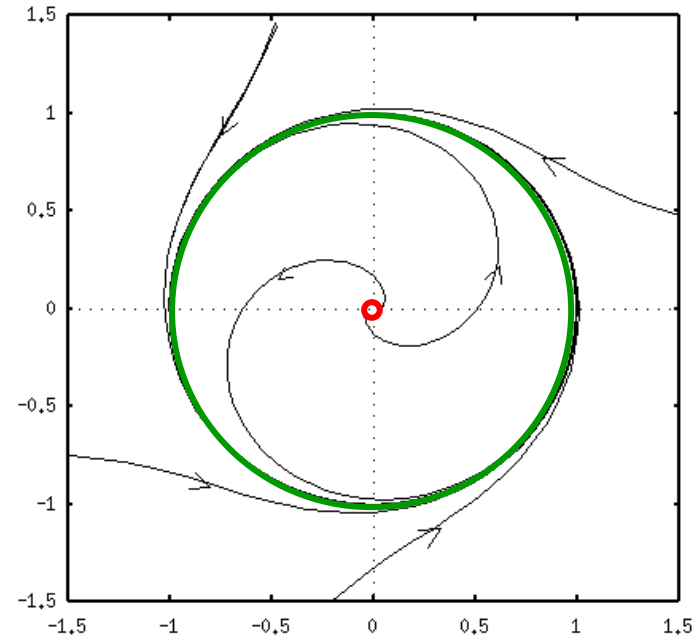
- A DS is structurally stable if small changes to its vector field lead to structurally similar phase portraits.

- Our example system is structurally stable.

Structural instability

$$\dot{r} = 0 \quad \dot{\theta} = 1$$
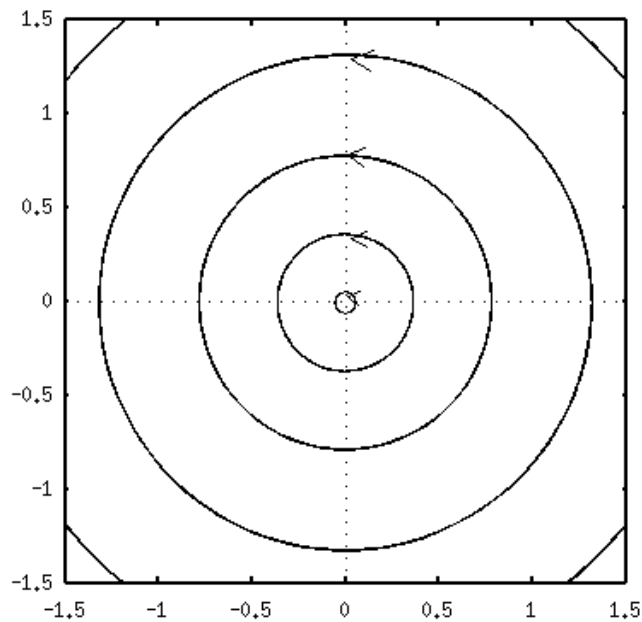
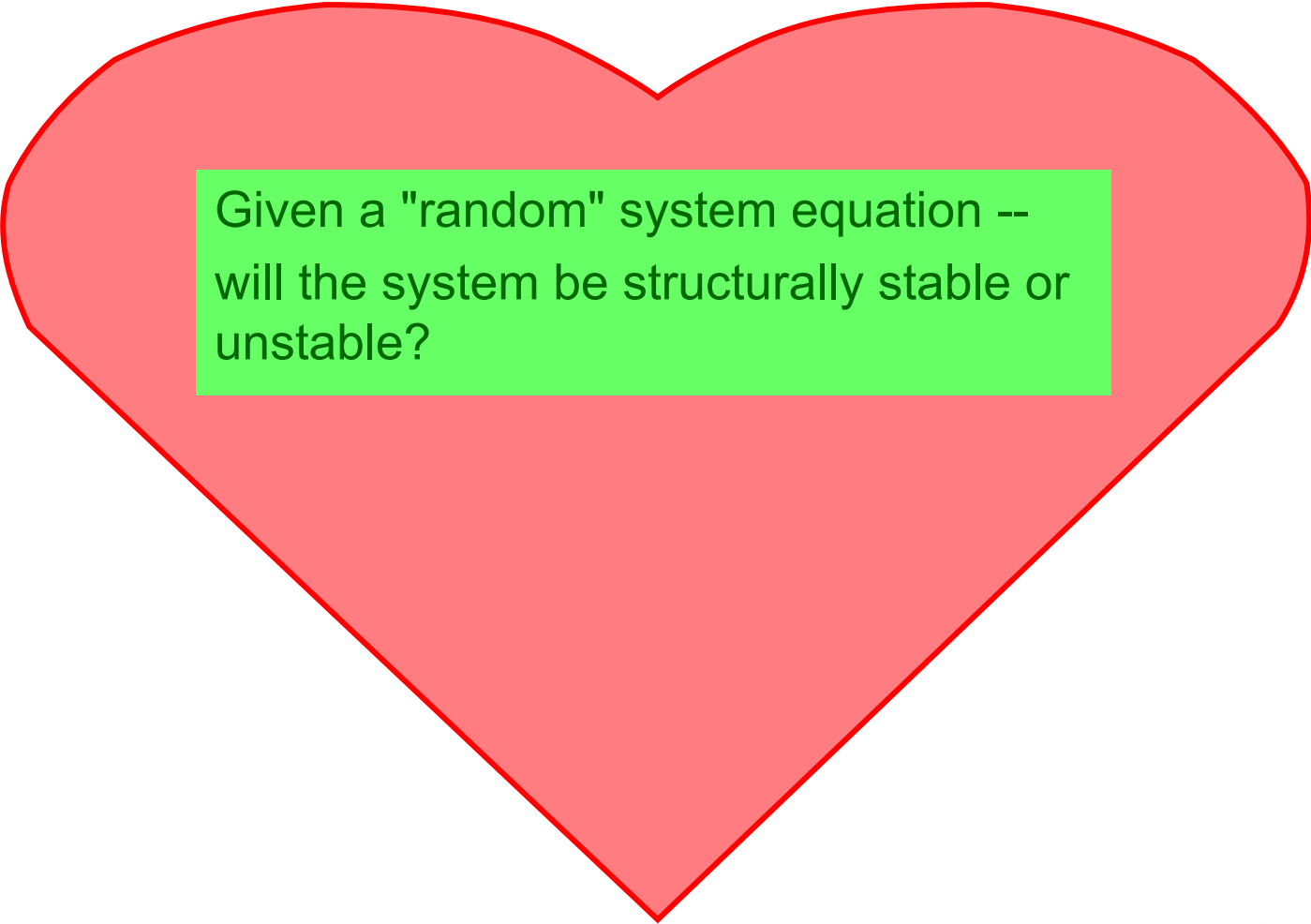$$\dot{r} = 0.01 \quad \dot{\theta} = 1$$

- This system is structurally not stable.
- Arbitrarily small changes in ist equations will make the circles "miss themselves on their return trip" -- giving spiral portraits instead (inbound or outbound)
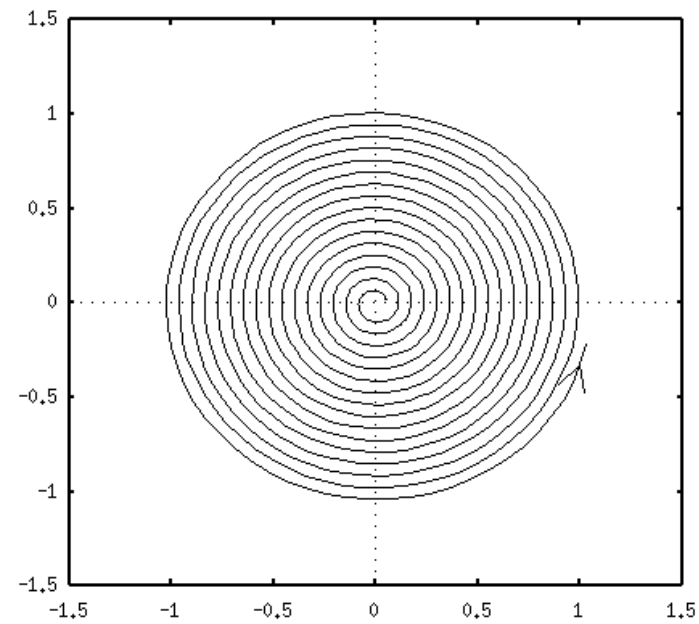
- This "spiral" phase portrait is structurally stable.

JACOBS UNIVERSITY

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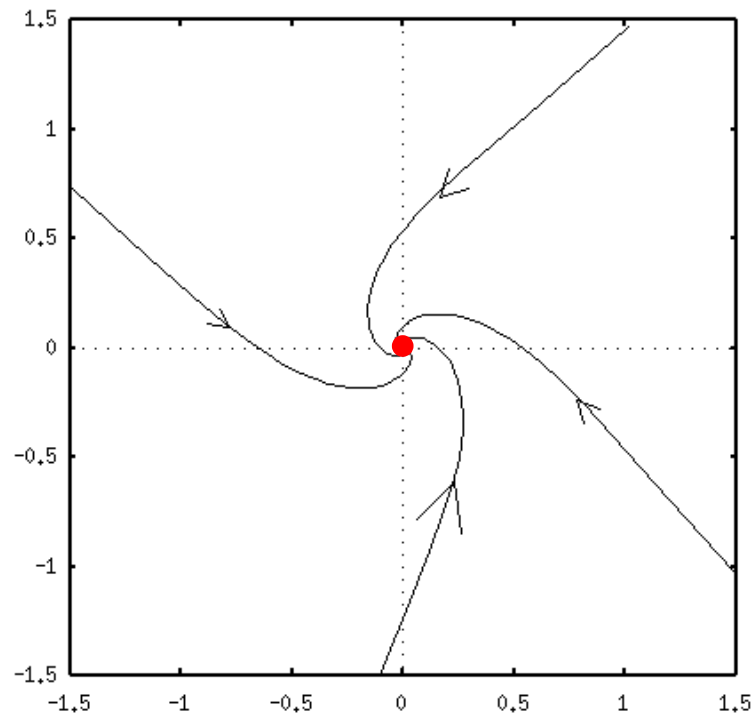
[Er] nahm darauf mein Buch, so ich eben bei mir hatte, und nachdem er sich in einen Schreiber verwandelt, schrieb er mir nachfolgende Worte darein: "Ich bin der Anfang und das End, und gelte an allen Orten. Magst dir selbst einbilden wie es einem jeden Ding ergangen, hernach einen Discurs daraus formirn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

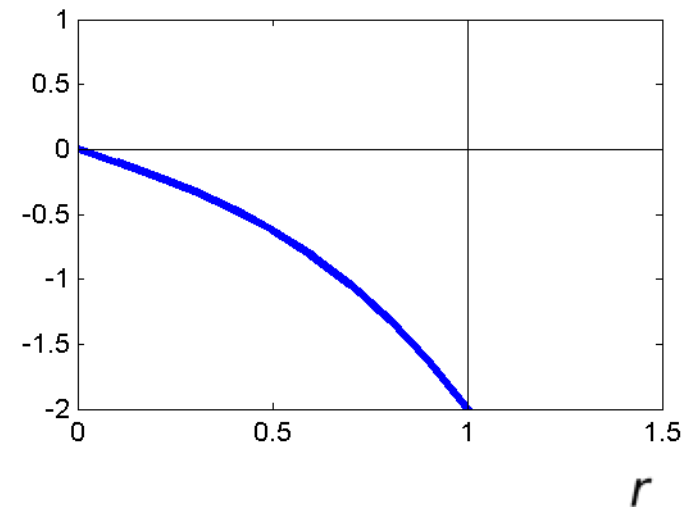*Grimmelshausen, Simplicius Simplicissimus, sechstes Buch, Kapitel 9*
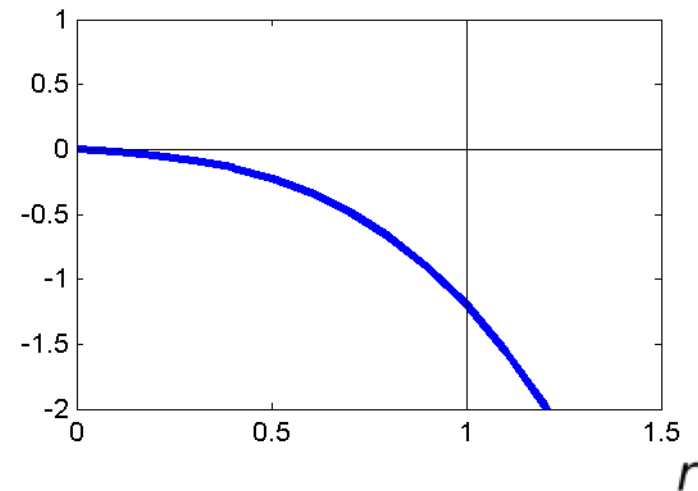
JACOBS
UNIVERSITY

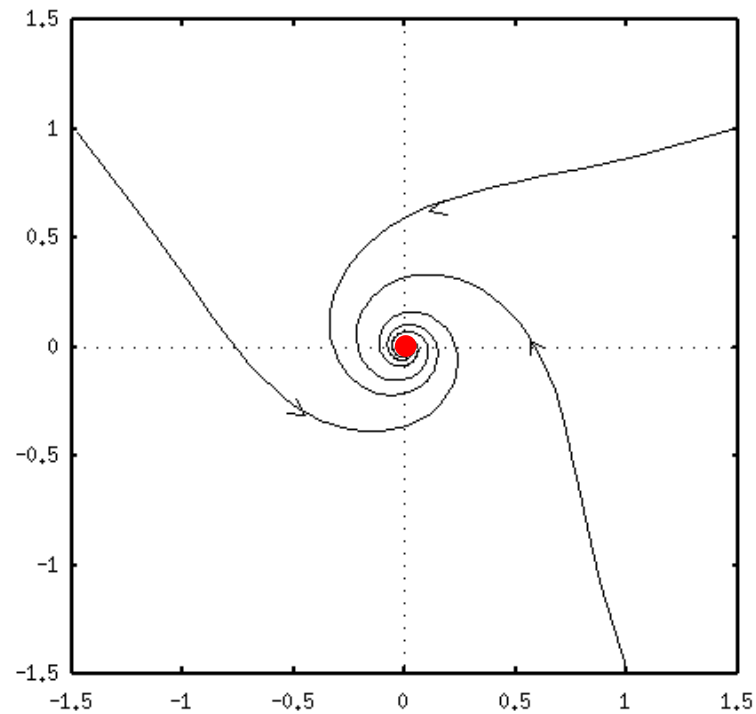# Parameter magic...

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

$$a = -1$$

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

JACOBS
UNIVERSITY

Parameter magic...

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

$$a = -0.2$$

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

Parameter magic...

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

$$a = 0$$

$$\dot r = -r^3 - 0\,r$$

JACOBS
UNIVERSITY

Parameter magic...

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

$$a = 0.1$$

$$\dot{r} = -r^3 + 0.1\,r$$

JACOBS
UNIVERSITY

Parameter magic...

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

$$a = 1$$

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

JACOBS
UNIVERSITY
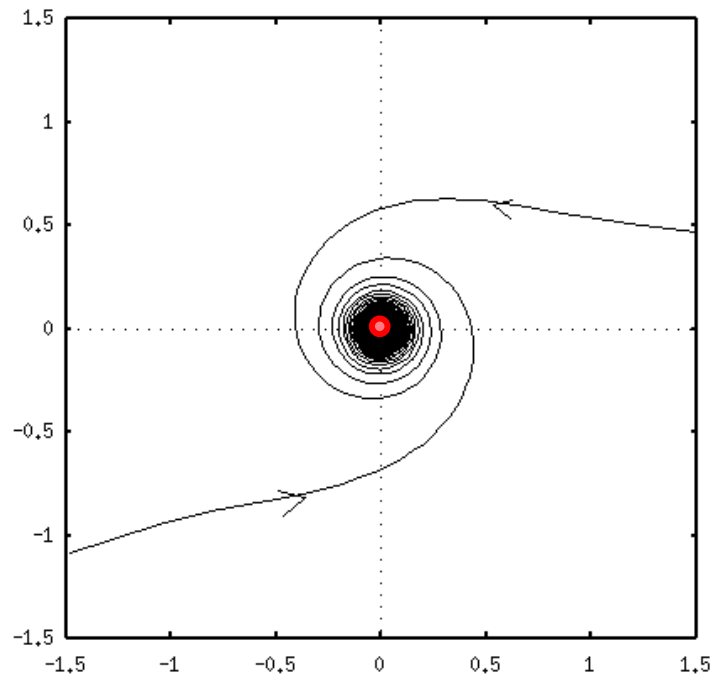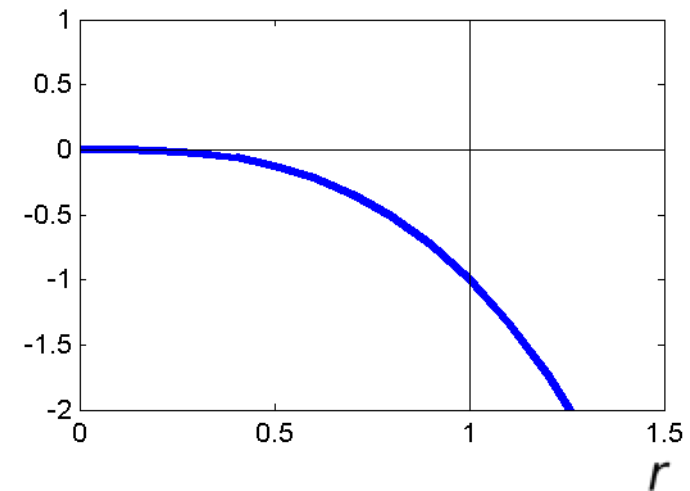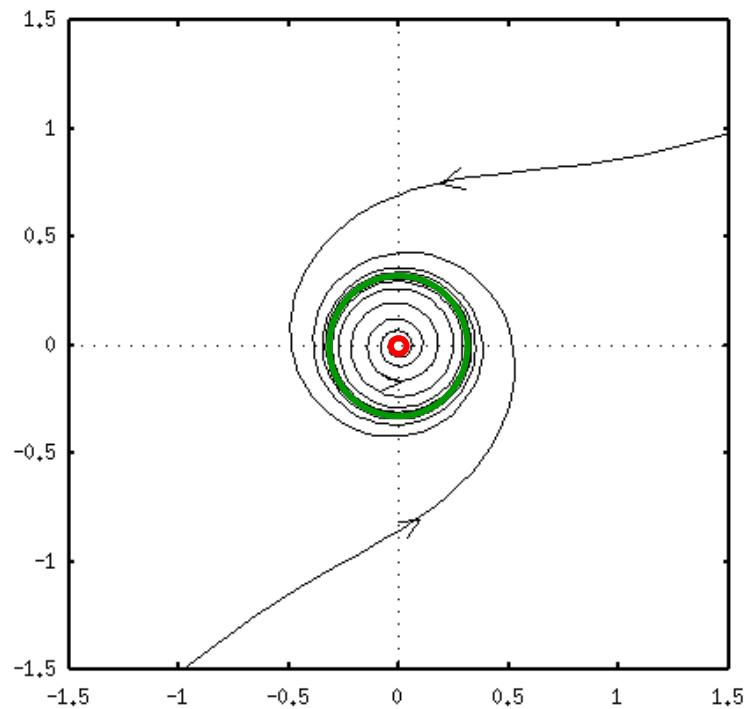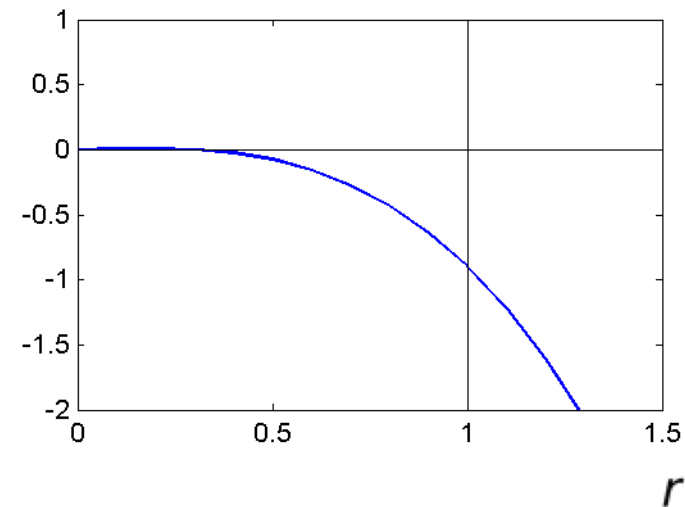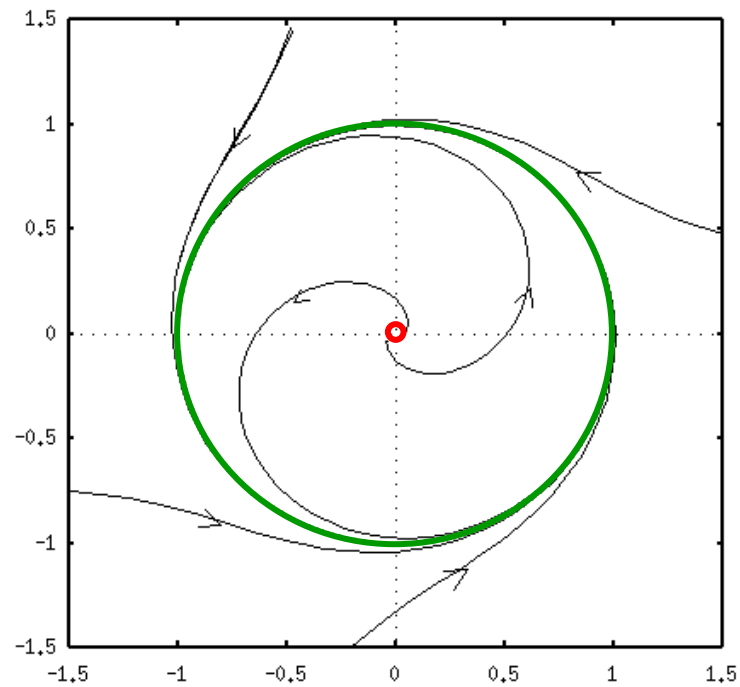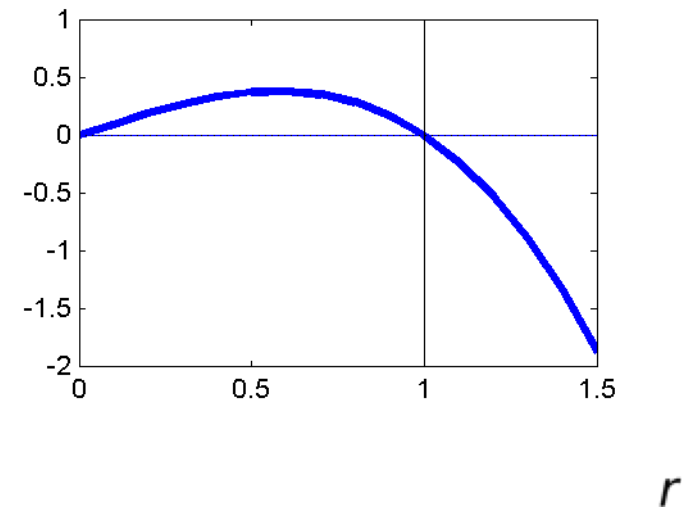
It's a Bifurcation.

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

| $a = -1$ | $a = -0.2$ | $a = 0$ | $a = 0.1$ | $a = 1$ |
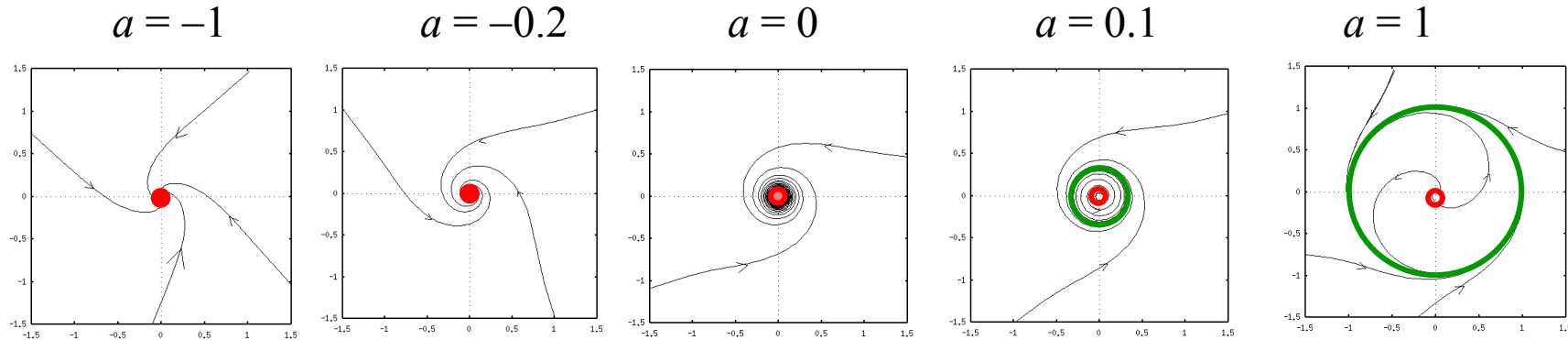
$a < 0$: Stable fixed point at origin (a point attractor)

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 ...

JACOBS UNIVERSITY

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 http://www.dynamicalsystems.org/tu/ .

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.

JACOBS
UNIVERSITY

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.
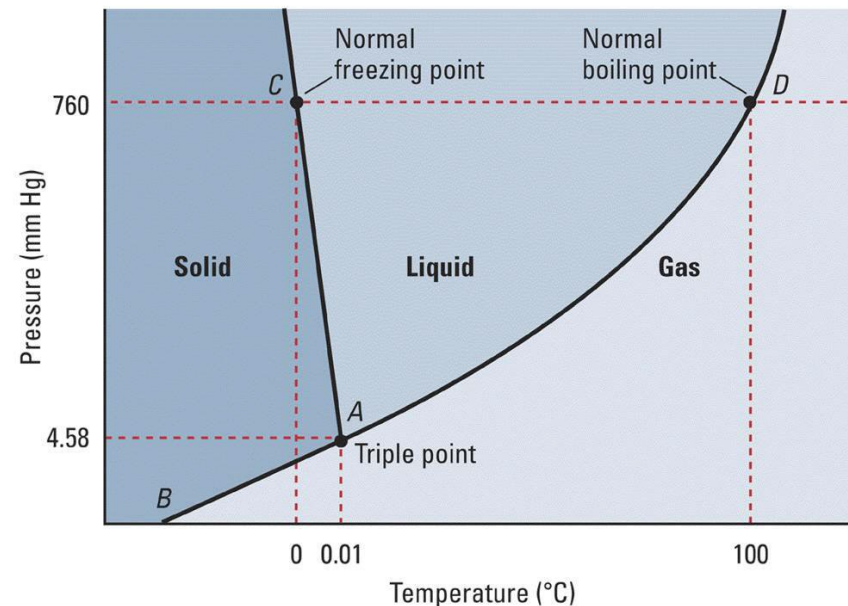
Phase transition diagram of water

- 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.

JACOBS UNIVERSITY

Literature / resource

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

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

JACOBS UNIVERSITY

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

(i) $\emptyset \in \mathcal{O}$,

(ii) $\mathcal{O}$ is closed under arbitrary unions,

(iii) $\mathcal{O}$ 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 $U \in \mathcal{O}$.

JACOBS
UNIVERSITY

Attractors defined in topological dynamics

Definition. Given: topological state space X, map $T: X \to X$. 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.

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

orbit of b

$\ldots$

$T(T(b))$

$T(b)$

A, U, B, b, X

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!

JACOBS UNIVERSITY
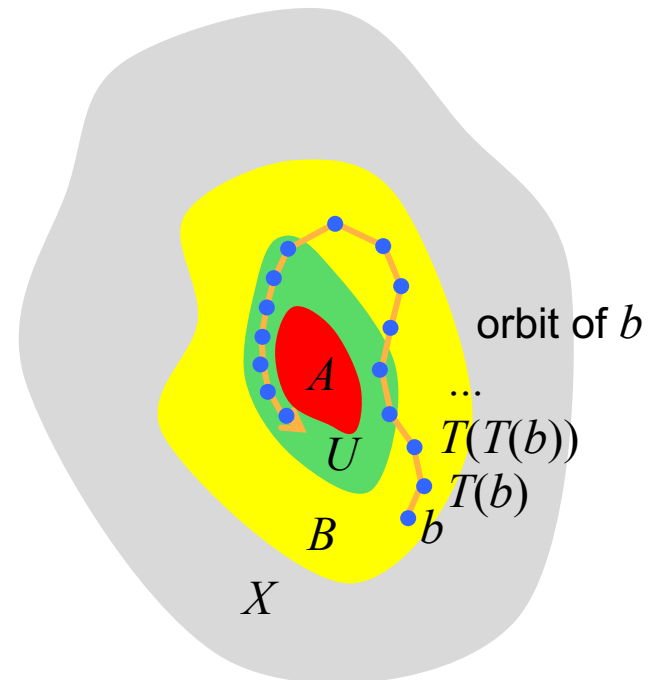
Example: cyclic attractors and their basins in a 16-cell CA

see detail
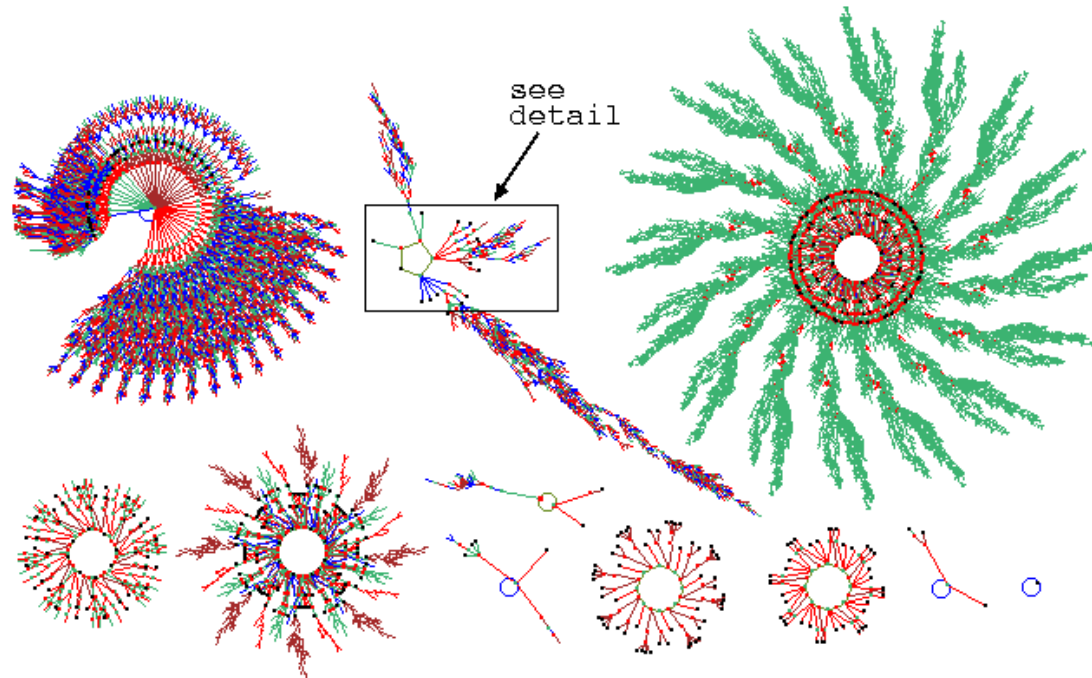
- State space X: binary 16-bit patterns
- CA transition rule yields $T: X \rightarrow X$
- discrete topology $O = \mathrm{Pot}(X)$

garden-of-Eden states

attractor cycle

transient trees and sub-trees

JACOBS UNIVERSITY
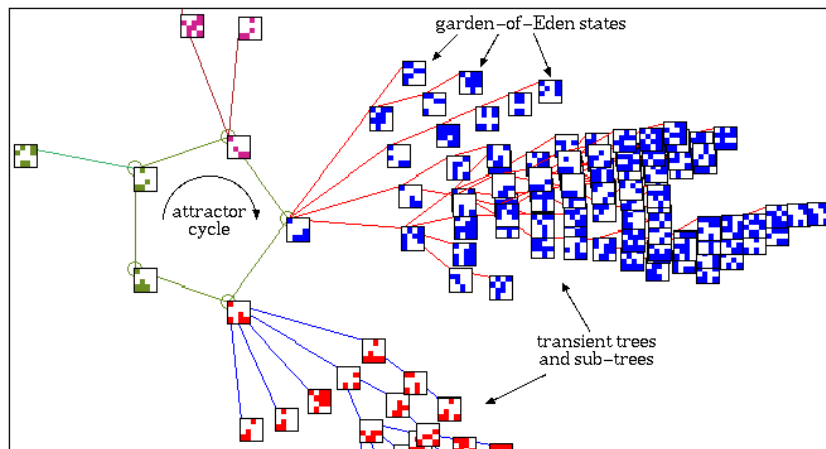
Example: a point attractor in a continuous 2-dim DS

fixed point attractor

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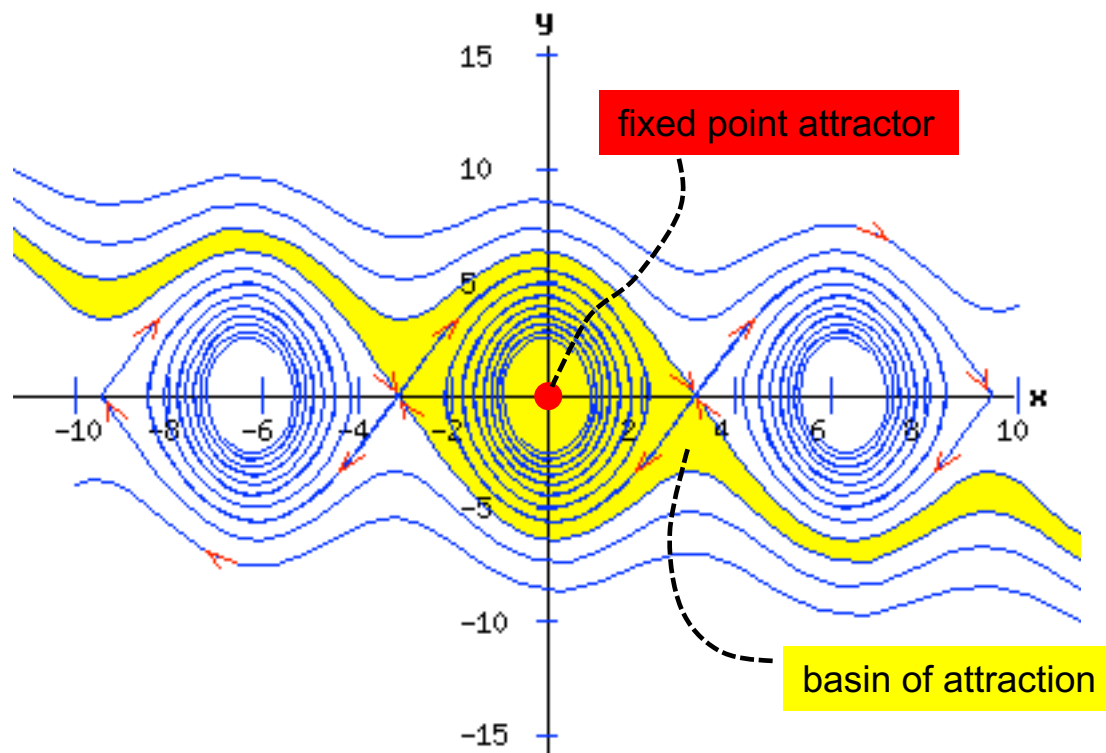
- State space X: $\mathbb{R}^2$
- continuous dynamics generated by

$$\dot{x} = y$$

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

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

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

JACOBS UNIVERSITY

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)

JACOBS UNIVERSITY

Attractors and symbols

The riddle

Neurons fire.

Humans think.

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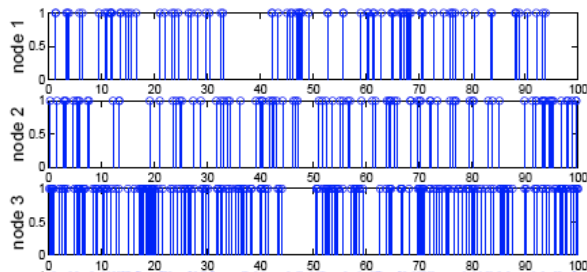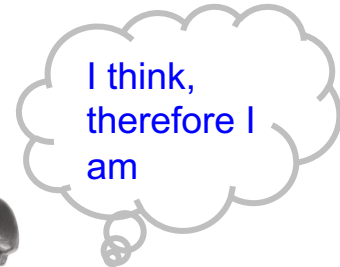
The golden riddle of neural-symbolic integration

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

- What are the neural correlates?

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

store.metmuseum.org, www.theguardian.com/education/2012/jun/11/gold-wedding-rings-trail-marital-disintegration

JACOBS UNIVERSITY

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

JACOBS UNIVERSITY

Literature

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

S.Bader, P.Hitzler: Dimensions of Neural-symbolic Integration – A Structured Survey. In S. N. Artëmov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb, and J. Woods, editors, We Will Show Them!, pages 167–194. College Publications, 2005. http://arxiv.org/pdf/cs/0511042.pdf

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

JACOBS UNIVERSITY

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 capture 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

JACOBS UNIVERSITY

Literature

References for the listed variations of attractor concepts can be found in

H. Jaeger (2014): Controlling Recurrent Neural Networks by Conceptors.
Jacobs University technical report Nr 31 (page 89f)
http://arxiv.org/abs/1403.3369

JACOBS
UNIVERSITY

Part 7: Non-Autonomous Dynamical Systems

JACOBS
UNIVERSITY

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

JACOBS
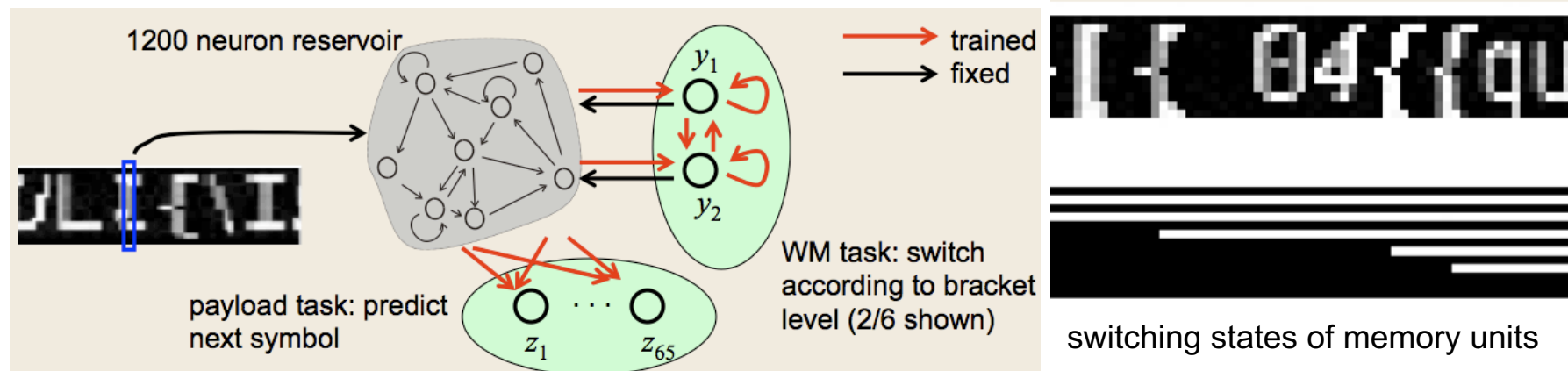UNIVERSITY

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: $T_n(\mathbf{x}(n)) = T(\mathbf{x}(n)) + \nu(n)$,
  - nonstationarity: T_n is intrinsically time-varying
- General theory of non-autonomous dynamical systems (NDS) abstracts away from specific sources of non-autonomy

JACOBS
UNIVERSITY

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

y_1

y_2

trained
fixed

payload task: predict next symbol

z_1 $\cdots$ z_{65}

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

switching states of memory units

JACOBS
UNIVERSITY

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

http://minds.jacobs-university.de/sites/default/files/uploads/papers/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.
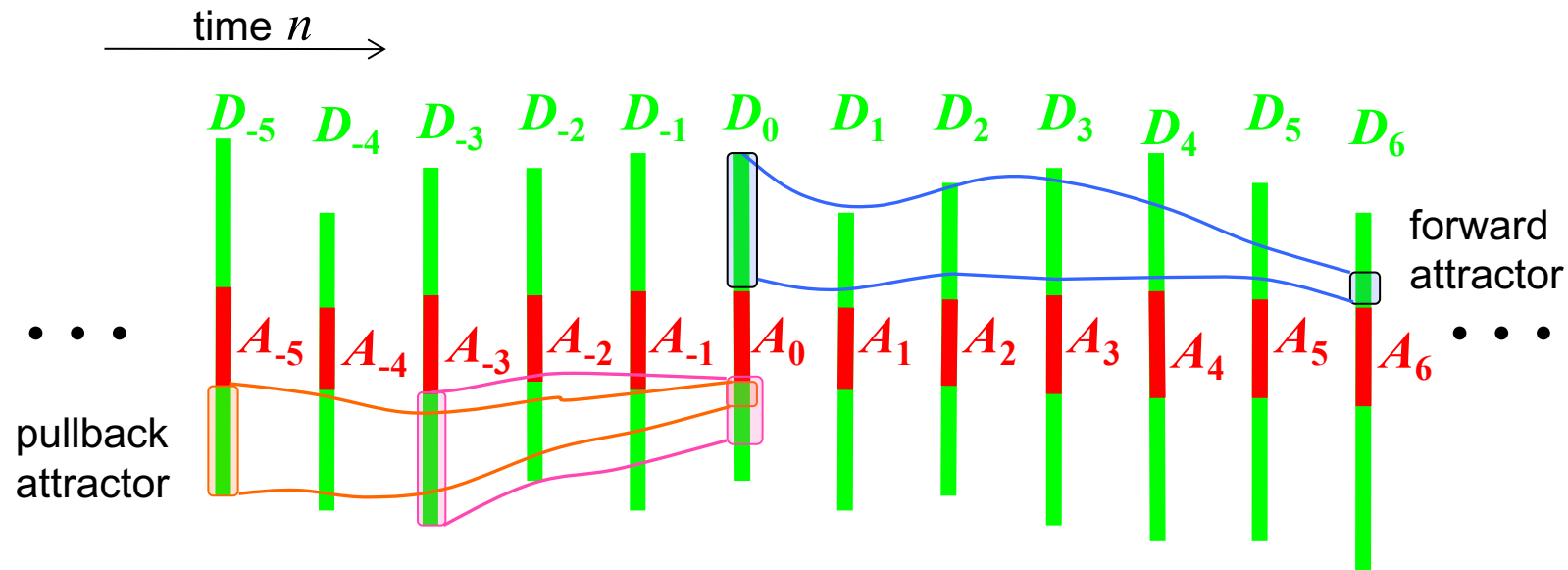
Specific assumptions

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

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

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

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

JACOBS
UNIVERSITY

Time-dependent attractors

- consider a time-varying set $(A_n)_{n \in \mathbb{Z}}$ 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.

JACOBS
UNIVERSITY

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.
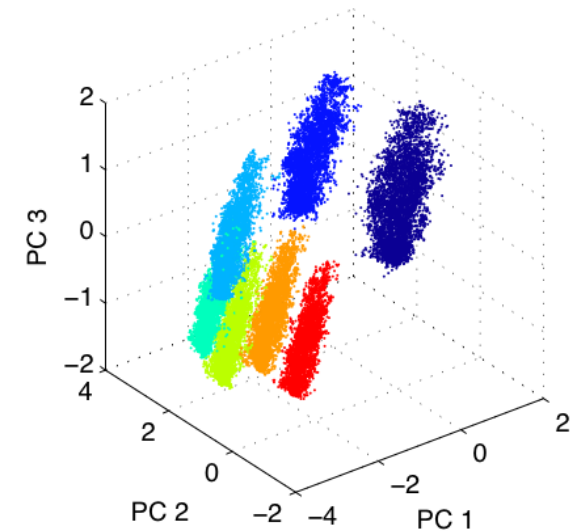
state space regions visited by working memory network when locked in different memory modes.

JACOBS
UNIVERSITY

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 wildly 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!

shovel bifurcation

JACOBS UNIVERSITY
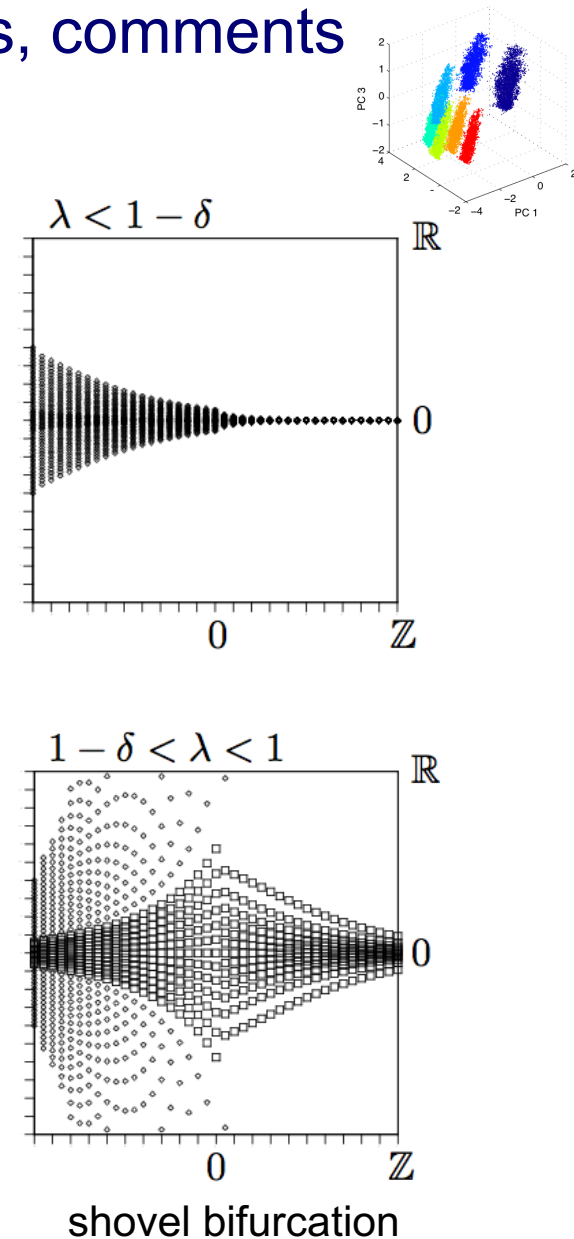
Literature

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

http://minds.jacobs-university.de/sites/default/files/uploads/papers/2499_ManjunathJaeger13a

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

JACOBS UNIVERSITY

Appendix: Themes not Covered in this Tutorial

JACOBS
UNIVERSITY

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
- time warping

JACOBS
UNIVERSITY