Simulation, Analysis, and Validation of Computational Models

— 6.a Noise in Dynamical Systems —

— 6.b Systems engineering —

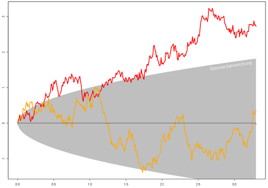
Lecturer: Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

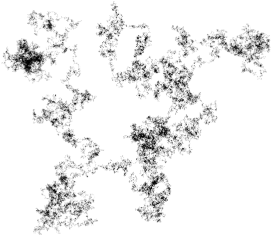
- Noise: Brownian motion, white and other noise
- System modelling and engineering
- Labs

Random walks: Brownian motion

- Brownian motion is the movement of small particles that are subject to many random kicks
- Particles are driven by noise: $\dot{x} = \xi$
- Percolation focuses on geometry, while here dynamics is crucial. Patterns are self-similar in both, but percolation clusters merge (fractal dimension at criticality: 1.75, in 2D), while being revisited in RW (fractal dimension: 2, in 2D)
- Applications of random walks include fluctuating stock prices, foraging pattern in animals, models for decision making, and exploration in machine learning.
- For low-level simulations of RWs, Monte-carlo methods are useful.



1D (Thomas Steiner, 2005)



2D (Purpy Pupple, 2011

Random walks: Wiener process

Brownian motion can be described mathematically by the Wiener process, which is defined by (the following holds with prob. 1)

- $W(t) \subseteq \mathbb{R}^d$ (state), $0 \leq t \in \mathbb{R}$ (time), $W(0) = 0$ (initialisation)
- $W(t)$ is subject to random independent identically distributed (i.i.d.) increments (whether $\pm\Delta t$ or continuously distributed)
- Increments are normally distributed $W(t) - W(s) \sim \mathcal{N}(0, t - s)$ for $t - s \gg 0$ (i.e. not considering the limit $t \rightarrow s$)

and has the following properties

- Realisations $W(t)$, $t \geq 0$ are continuous, nowhere differentiable
- If the process is stopped and mirrored (on a parallel of the t-axis) is still a perfect Wiener process
- $D = 2$: Gets arbitrarily close to all points in the plane, $D_{\text{frac}} = 2$
- $D \geq 3$: Arbitrarily large holes left

White noise

- White noise $\xi(t)$ stands in for the *supposed* derivative dW_t/dt of the Wiener process (the kicks underlying Brownian motion)
- However, dW_t/dt does not exist because the Wiener process is nowhere differentiable. Solutions:
  - Mathematical theory of distributions or
  - Noise term is a derivative of a differentiable (e.g. Fourier) interpolation of the Wiener process.
- Formally, the spectrum of the while noise does not decay with increasing frequency
  - for a physical process the total energy would be infinite
  - for a computational process the resolution would be infinite

  so WN is an idealisation, practically has cut-off frequency
- In other words: Noise is not a very intuitive concept, but that's the idea, but in many cases we know its statistical properties.
- White noise: $\langle \xi(t) \rangle = 0$, $\left\langle \xi^2(t) \right\rangle = 1$, $\langle \xi(t)\, \xi(s) \rangle = 0 \ \forall t \neq s$

Linear systems with additive noise

Standard 1D system with added noise term

$$\dot{x}(t) = c\,x(t) + \xi(t)$$

Formal solution for $x(0) = x_0$:

$$x(t) = x_0 \exp(c\,t) + x_0 \int_0^t \exp(c(t-s))\,\xi(s)\,ds$$

$c < 0$: stable, forgets initial condition, but noise keeps the system "alive" (Ornstein–Uhlenbeck process)

$c = 0$: Wiener process, diverges very slowly (see above)

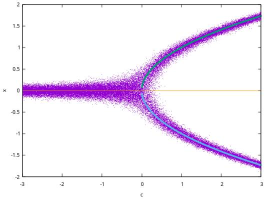
$c > 0$: exponential + exponentially accumulated random growth

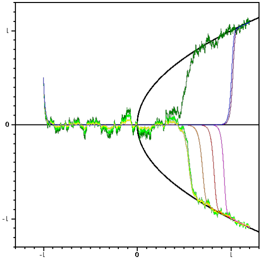
Phase transitions with noise

$$\dot{x} = cx - x^3 + \xi, \quad x(0) = 0$$

- Noisy variant of pitchfork bifurcation
- Noise effect maximal at $c = 0$, but can be non-trivial
- For slowly increasing c super-criticality can occur even at vanishing noise ($\lim_{\sigma \to 0} \sigma\xi$) (identical noise realisations).

$\sigma = 0$, $\sigma = 2 \cdot 10^{-10}$, $\sigma = 2 \cdot 10^{-8}$, $\sigma = 2 \cdot 10^{-6}$, $\sigma = 2 \cdot 10^{-4}$, $\sigma = 0.02$, $\sigma = 0.04$, $\sigma = 0.06$.





Nils Berglund, Inst. Denis Poisson

Stable distributions

- Central limit theorem guaranties normal distribution of a normalised sum of many random variables, if the variance of each is bounded.
- If the variance of the components is not bounded: Lévy α-stable distributions $\alpha \in (0, 2]$ (shape parameter)
  - $\alpha = 2$: normal distribution (limit case)
  - $1 < \alpha < 2$: expectation exists, but variance does not exit
  - $0 < \alpha \leq 1$: expectation does not exist
- A stable distribution means that if X_1 and X_2 are independent with the same distribution, then for any constants $a > 0$ and $b > 0$ also the random variable $aX_1 + bX_2$ has the same distribution (up to linear scaling and possibly linear shift).
- The *normal distribution* and α-stable distributions are generic for continuous cases.
- For discrete cases the Poisson distribution is the limit distribution of discrete-stable distribution.
- For the occurrence of distributions that are not stable nor normal an explanation should be possible.

Random number generators

- Typically pseudo-random numbers are used
  - good speed and evenness in generation
  - reproducibility (fixed seed)
- Python's "import random" provides Mersenne Twister (1997) with a period of $2^{19937} - 1$ and equi-distribution of up to 623 dimensions (for 32 bit values).
- random.shuffle(x) can improve "randomness"
- How to generate other than homogeneous distributions?
  - many distributions predefined
  - rejection method or functional
- Standard pseudo-random generators should not be used for security purposes
- In cryptographically secure pseudo-random number generators knowledge of seed does not provide significant advantage. They may get outdated soon, or may be compromised.

Conclusion: Noise in dynamical systems

- Noise in linear systems counteracts stability, i.e. it can become a problem in weakly stable systems.
- Noise in nonlinear systems can have non-trivial effects, e.g. vanishing noise ($\lim_{\sigma \to 0} \sigma \xi$) may not reproduce the deterministic case.
- The (generalised) central limit theorem states that the normalised sum of many random variable has a normal distribution or an α-stable distribution.

Complex systems

Characterisation of complex systems?

- open systems
- nested
- various types of interaction

Example: Physics Nobel prize 2021

- Syukuro Manabe and Klaus Hasselmann "for the physical modelling of Earth's climate, quantifying variability and reliably predicting global warming"
- Giorgio Parisi "for the discovery of the interplay of disorder and fluctuations in physical systems from atomic to planetary scales"

Dynamical systems modelling

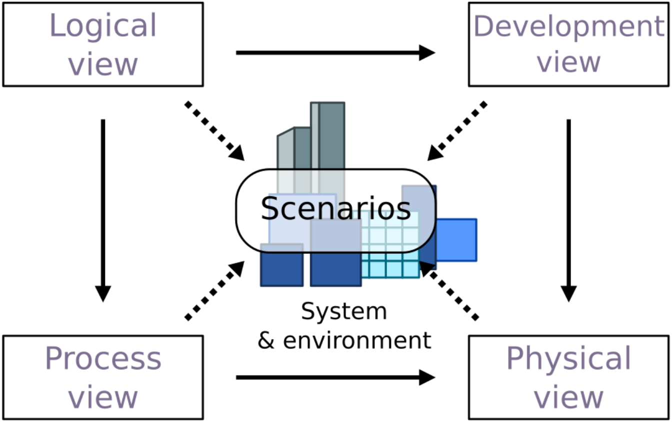
- System identification
- Functional modeling
- Systems architecture
- Complex systems

Analysis and synthesis of systems

- "to take apart" and "to put together"
- Method or discipline?
- Includes software systems as well as physical systems
- Reduction of complexity (Separation of concerns, encapsulation, ...)
- Prototyping
- Adding personnel to a software project that is behind schedule delays it even longer.

Fred Brooks (1975) The Mythical Man-Month: Essays on Software Engineering

4+1 architectural view model



sFile:4+1 Architectural View Model.jpg by User:Mdd CC BY-SA 3.0

Systems engineering: Features

- Interdisciplinary: science and engineering, design, development, project management, logistics, stake holder domain, risk management
- Features:
  - Products: design of physical systems consisting of hardware and software
  - Enterprises: combinations of organizations as systems
  - Services: civil infrastructure

Systems engineering: Early example

- US space shuttle (1981): Disasters in 1986 (Challenger) and 2003 (Columbia) reliability 98.5%, programme ended in 2011.
- EU Ariane rocket family
  - Ariane 1 and 2 $\approx$ 80% reliability (1974-1989)
  - Ariane 3 and 4 $\approx$ 100% (single failures per subtype) 1984-2003 (redesign based on systems engineering)
  - Ariane 5: 81% $\rightarrow$ 100% Ariane 6 (today): 81% (9 out of 11 successful)

Systems engineering

Management process interacts with technical process (see UG SDP course) that consisting of

- Assessing available information
- Defining effectiveness measures
- Creating behavior model
- Creating structure model
- Performing trade-off analysis
- Creating sequential build and test plan

Oliver, D.W. e.a. (1997). Engineering Complex Systems with Models and Objects. McGraw-Hill.

"Waterfall" model

Development activities as linear sequential phases

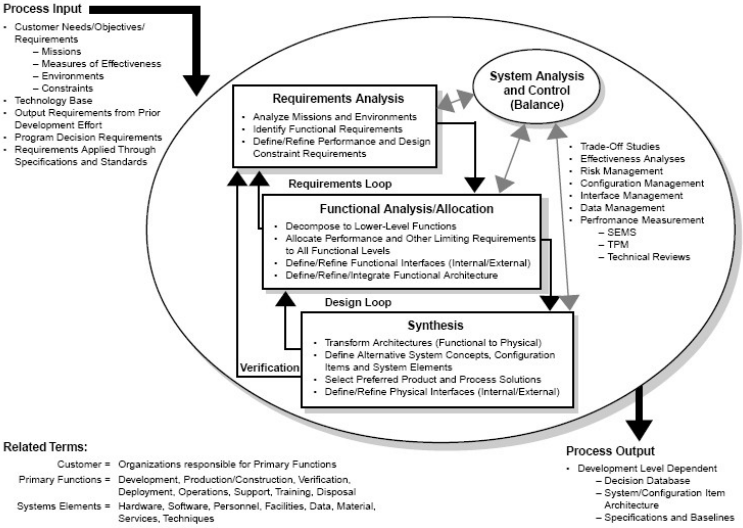
- System and software requirements: Product requirements document
- Analysis: Resulting in models, schema, and business rules
- Design: Resulting in the software architecture
- Coding: Development, proving, and integration of software
- Testing: Systematic discovery and debugging of defects
- Operations: Installation, migration, support, and maintenance of complete systems

Petersen, Wohlin, Baca, Dejan (2009) The Waterfall Model in Large-Scale Development. In: Product-Focused Software Process Improvement, Springer, pp. 386–400.

Lineberger, Rob (2024) Inheriting Agile: The IT Practitioner's Guide to Managing Software Development in a Post-Agile World. Sandprint Press. p. 37.

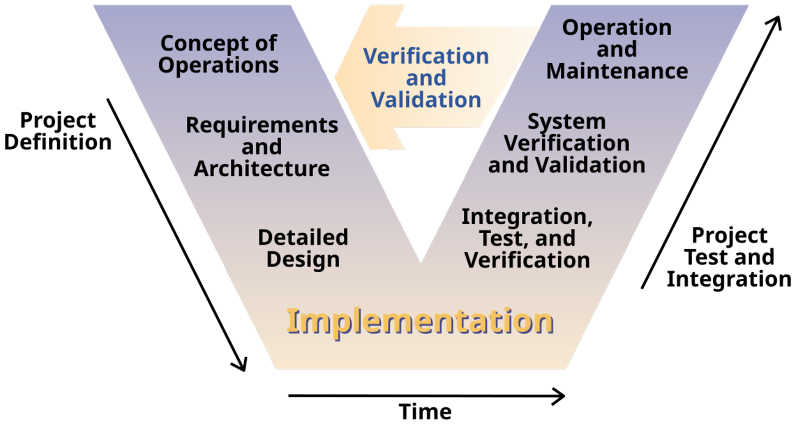
Criticism: Testing happens only in the end; risky as requirements are usually not fully known in advance

W Royce (1970) Managing the Development of Large Software Systems. Proc. IEEE WESCON, 26: 1–9



Systems Engineering Fundamentals. Defense Acquisition University Press (public domain)

V-model



L. Osborne e. a. (2005) Clarus Concept of Operations. FHWA-JPO-05-072, (Redrawn by Slashme).

Conclusion: Systems engineering

- Overarching system development models help to structure tasks by specifying what to consider in what order
- They are evidently useful in heterogeneous teams of experts
- They don't solve technical problems nor replace critical thinking, overview of side effects, and personal responsibility
- Today, they are used to define the approach and to provide guidance for subprojects

Next topics

- Simulations
- MAS
- Networks

Lab 1: Systems analysis

- Linear Systems
- Routes to Chaos
- Phase transition
- Noise

Lab 2: Systems modelling

- System modelling
- Simulations
- PyBullet

Lab 3: Integration of data

- Autograd
- Towards PINN