Simulation, Analysis, and Validation of Computational Models

— Linear Systems —

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

Overview

- Linear systems
- Examples
- ODE
- Outlook

Last time: Exponential growth/decay in discrete time

For $\mathcal{S} = (0, \infty)$, $\mathcal{I} = \{0, 1, 2, \dots\} = \mathbb{N}_0$, consider the dynamical system:

$$S_{t+1} = aS_t,$$

where $a > 0$ is a constant growth factor. Given S_0, we find*

$$S_t = a^t S_0$$

If the sequence is shifted $\tilde{S}_0 = S_{t_1}$ then

$$\tilde{S}_{t_2} = a^{t_2}\tilde{S}_0 = a^{t_2}a^{t_1}S_0 = a^{t_2+t_1}S_0 = S_{t_2+t_1}.$$

Asymptotic behaviour

$$\lim_{t\to\infty} S_t = \begin{cases} 0 & a < 1 \\ S_0 & a = 1 \\ \infty & a > 1 \end{cases}$$

Exponential growth/decay in continuous time

For $\mathcal{S} = (0, \infty)$, $\mathcal{I} = [0, \infty) = \mathbb{R}_0^+$, consider the dynamical system:

$$S_{t+1} = aS_t \qquad (1)$$

In a continuous system only small steps can be made, so consider

$$S_{t+1} - S_t = aS_t - S_t,$$

$$\frac{S_{t+\Delta t} - S_t}{\Delta t} = aS_t - S_t, \text{ for } \Delta t = 1$$

Choice of the time unit is arbitrary, so it can as well be small:

$$\frac{S_{t+\Delta t} - S_t}{\Delta t} = aS_t - S_t, \text{ for } \Delta t > 0$$

Two steps (not much has changed from the discrete case):

1. Calculate change: $\Delta S_t = (a-1)\, S_t$, where $\Delta S_t = \frac{S_{t+\Delta t} - S_t}{\Delta t}$
2. Update: $S_{t+\Delta t} = S_t + \Delta S_t$, and increment time by Δt

Exponential growth in continuous time

Is it feasible to consider arbitrary short times, i.e. $\Delta t \to 0$?

$$\lim_{\Delta t \to 0} \frac{S_{t+\Delta t} - S_t}{\Delta t} = \frac{dS_t}{dt} = (a-1)\, S_t,$$

We will now write c instead of $a-1$, and $x(t)$ instead of S_t. Does the equation (A dot over a variable denotes the time derivative.)

$$\frac{dx(t)}{dt} = cx(t) \quad \text{or} \quad \dot{x}(t) = cx(t) \tag{2}$$

describe exponential growth? To check, we can simulate (later) or calculate analytically (see below).

What is the meaning of $\dot{x}(t) = cx(t)$?

- Causal: Temporal change depends on the function value.
- Epistemic: Given $x(0)$, Eq. 2 characterises a function $x(t)$.
- Mathematical: Find function $x(t)$, given $x(0)$ and Eq. 2.
- Algorithmic: Find a good approximation of $x(t)$.

Exponential growth in continuous time (analytical)

Solving the equation $\dot{x}(t) = cx(t)$ analytically:

$$\frac{dx(t)}{dt} = cx(t)$$

or, if[1] $x(t) \neq 0$,

$$\frac{1}{x(t)} \frac{dx(t)}{dt} = c$$

Integrate both sides

$$\int_{t_0}^{t_1} \frac{1}{x(t)} \frac{dx(t)}{dt} dt = \int_{t_0}^{t_1} c \, dt$$

Substitute $x = x(t)$ where $dx = \frac{dx(t)}{dt} dt$ ("cancel the dt") with $x_0 = x(t_0)$ and $x_1 = x(t_1)$

$$\int_{x_0}^{x_1} \frac{1}{x} dx = \int_{t_0}^{t_1} c \, dt$$

[1] Assumption to be checked later.

Exponential growth in continuous time (analytical)

Solve the integrals

$$
\begin{aligned}
\int_{x_0}^{x_1} \frac{1}{x} dx &= \int_{t_0}^{t_1} c dt \\
\log(x_1) - \log(x_0) &= c(t_1 - t_0) \\
\log\left(\frac{x_1}{x_0}\right) &= c(t_1 - t_0) \\
x_1 &= x_0 \exp(c(t_1 - t_0))
\end{aligned}
$$

For $t_0 = 0$, $t_1 = t$, and $x_1 = x(t)$, we find the solution:

$$x(t) = x_0 \exp(c\, t) \tag{3}$$

$$\lim_{t \to \infty} x(t) = \begin{cases} 0 & c < 0 \\ x_0 & c = 0 \\ \infty & c > 0 \end{cases}$$

Properties of the exponential function imply semigroup property, see last lecture

Exponential growth in continuous time: Remarks

Discrete (1) and continuous (2) systems appear similar, however

- Comparing $x_0 \exp(ct) = x_0 (\exp c)^t$ and $S_0 a^t$, we find $a = \exp(c)$,
- $c = a - 1$ is a good approximation only for $c \approx 0$, $a \approx 1$, i.e. the size of the time step does not matter, if not much happens within a time step (what "happens" depends on the time step with which we have tampered, see slide 5).
- c can below -1, while $a > 0$, in all these cases the solution does not cross 0 from either side (see assumption above).
- If we start with $x(0) = 0$ or with $S_0 = 0$, nothing will happen.
- Similar to continuous compounding of interest
- Was the transition to the continuous case worth the effort, when we will consider simulated systems anyway?

Linear Dynamics in 2D for continuous time

$x = (-\infty, \infty)^2$, $\mathcal{I} = [0, \infty) = \mathbb{R}_0^+$, consider the dynamical system:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = B \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, \quad (4)$$

where B is a constant matrix. Given $x(0) = (x_1(0), x_2(1))^\top$, we are going to solve (4). In case $b_{12} = b_{21} = 0$, this is easy (see above).

Idea: Find rotation matrix $R^\top$ to rotate B into diagonal form $D \rightarrow$ easy solution of decoupled system $\rightarrow$ rotate back.

$$\begin{aligned} \dot{x} &= Bx \\ \underbrace{R^\top \dot{x}}_{\dot{y}} &= \underbrace{R^\top B R}_{D} \underbrace{R^\top x}_{y} \end{aligned}$$

where R is an orthogonal (rotation) matrix, $RR^\top = 1$ (unit matrix). The matrix R will be obtained from the eigenvectors of B.

Eigen-decomposition: Assume we can find 2D vectors ρ_i, $i \in \{1, 2\}$ with $\lambda_i \rho_i = B\rho_i$ for some λ_i, then $B = \sum_{i=1}^{2} \lambda_i \rho_i \rho_i^\top$.

Linear Dynamics in 2D for continuous time

The columns of $R = (\rho_1, \rho_2)$ are eigenvectors of $B = \sum_{i=1}^{2} \lambda_i \rho_i \rho_i^\top$:

$$
\begin{aligned}
R^\top B R &= (\rho_1, \rho_2)^\top \sum_{i=1}^{2} \lambda_i \rho_i \rho_i^\top (\rho_1, \rho_2) \\
&= (\rho_1, \rho_2)^\top \lambda_1 \rho_1 \rho_1^\top (\rho_1, \rho_2) + (\rho_1, \rho_2)^\top \lambda_2 \rho_2 \rho_2^\top (\rho_1, \rho_2) \\
&= \lambda_1 (\rho_1, 0)^\top \rho_1 \rho_1^\top (\rho_1, 0) + \lambda_2 (0, \rho_2)^\top \rho_2 \rho_2^\top (0, \rho_2) \\
&= \lambda_1 \begin{pmatrix} \rho_1^\top \rho_1 \rho_1^\top \rho_1 & 0 \\ 0 & 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 0 & 0 \\ 0 & \rho_2^\top \rho_2 \rho_2^\top \rho_2 \end{pmatrix} \\
&= \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}
\end{aligned}
$$

because eigenvectors are normalised $\rho_1^\top \rho_1 = 1$, $\rho_2^\top \rho_2 = 1$, and orthogonal $\rho_1^\top \rho_2 = \rho_2^\top \rho_1 = 0$.

For this reason, also R is orthogonal, and it is easy to show that $R^\top = R^{-1}$, so it doesn't matter whether we use $R^\top B R$ or $R^{-1} B R$.

Linear Dynamics in 2D for continuous time

Putting things back together: Get solution of

$$\dot{y} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} y$$

simply by stacking solutions of Eq. 3 (above):

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} y_1(0)\exp(\lambda_1 t) \\ y_2(0)\exp(\lambda_2 t) \end{pmatrix}$$

where $y = R^\top x$, i.e. $x = R\, y$.

We transform the given initial conditions $x(0)$ to $y(0) = R^\top x(0)$, and obtain the solution:

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = R \begin{pmatrix} y_1(0)\exp(\lambda_1 t) \\ y_2(0)\exp(\lambda_2 t) \end{pmatrix}$$

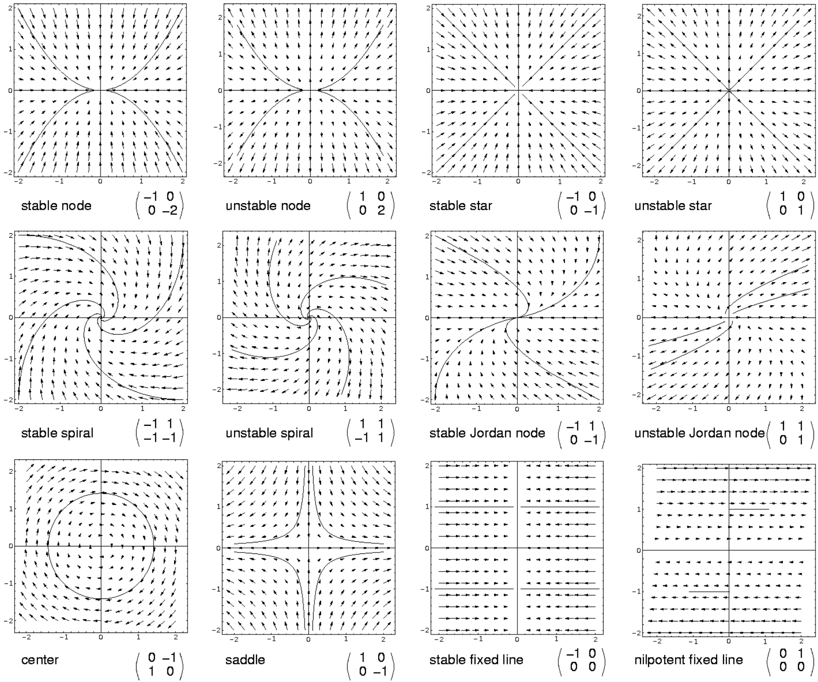
Linear Dynamics in 2D for continuous time: Remarks

- Works in an analogous way also for higher dimensions
- What if λ_i are complex? Use Euler formula for $k = 1, 2$:

$$\begin{aligned} y_k(t) &= (u_k(0) \pm i v_k(0)) \exp((z_k \pm i w_k) t) \\ &= (u_k(0) \pm i v_k(0)) \, y_k(0) \exp((z) t) (\cos wt \pm i \sin wt), \end{aligned}$$

"$\pm$" because for real matrices the non-real eigenvalues are pairs. Eigenvector are then also complex, and thus also initial conditions.

- What if the matrix B is not diagonalisable, i.e. $BB^\top \neq B^\top B$?
- Next slide shows asymptotic behaviours for almost all different cases. *Phase plot*: x space and vector of changes $\dot{x}$ (small arrow).



stable node $\begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$
unstable node $\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$
stable star $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$
unstable star $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
stable spiral $\begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix}$
unstable spiral $\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$
stable Jordan node $\begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix}$
unstable Jordan node $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$
center $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$
saddle $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
stable fixed line $\begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$
nilpotent fixed line $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$

Stability and control

- The system is stable, if all real parts of eigenvalues are negative.
- The system is unstable, if one eigenvalue has positive real part.
- Non-zero imaginary parts in any eigenvalues lead to oscillations or rotational effects.
- Control theory aims at moving or placing eigenvalues to the negative side (and to reduce oscillations).

Is this really all?

No.

- Dynamical variables can occur in non-linear functions.
- In quasi-linear cases, parameter matrix can change over time (including number of relevant eigenvalues).
- Derivatives w.r.t. different quantities (partial differential equations)
- Noise can affect state measurements or even perturb the parameters.
- Delays before a state measurement causes a change of state (control!) or there can be various delays.
- Non-standard solutions are possible (sliding mode).

Yes (almost).

- A simulation can appear realistic, if dynamic matrix changes in a suitable way (and collisions are treated separately).

How to make things move?

- Newton's 2nd law $F = m\,\ddot{x}$ (force = change of momentum)
- To control the dynamics, add forces!

$$a\ddot{x}\,(t) + b\,\dot{x}\,(t) + c\,x\,(t) = F\,(t)$$

- Practically, the coefficients a, b, c need to have the correct units, as the equation is all about acceleration [length/time^2].
- Higher-order time derivatives in linear systems can be treated like multi-dimensional linear (first-order) systems.

Example: Free fall in a gravitational field

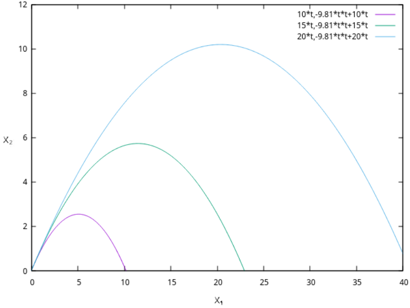
Gravitational acceleration $g = 9.81 \frac{m}{s^2}$ towards -x_2, no air resistance

$$\ddot{x}(t) = \begin{pmatrix} 0 \\ -g \end{pmatrix}$$

Integrate twice:

$$\dot{x}(t) = \begin{pmatrix} c_1 \\ -gt + c_2 \end{pmatrix}$$

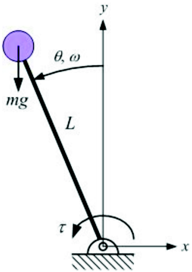
$$x(t) = \begin{pmatrix} c_1 t + d_1 \\ -gt^2 + c_2 t + d_2 \end{pmatrix}$$



To determine the integration constants, assume $x(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$,

$$v(0) = \dot{x}(0) = \begin{pmatrix} v_0 \\ v_0 \end{pmatrix} \Rightarrow c_1 = c_2 = v_0 \text{ and } d_1 = d_2 = 0$$

Example: Upright pendulum (higher-order system)



Movement determined by impulse, friction, potential energy, torque [ignoring the details]:

$$I\ddot{\theta} + k\dot{\theta} + Lmg\sin\theta = \tau$$

Known are current state $\dot{\theta}(t)$, $\theta(t)$ and input τ,

$$\ddot{\theta} = -\frac{k}{I}\dot{\theta} - \frac{Lmg}{I}\sin\theta + \frac{\tau}{I}$$

Linearisation ($\sin(x) \approx x$ if $x \approx 0$ or $x \approx \pi$)

$$\ddot{\theta} = -\frac{k}{I}\dot{\theta} - \frac{Lmg}{I}\theta + \tilde{\tau}$$

Formally we set $\omega = \dot{\theta} = \theta_1$ and $\theta = \theta_2$, i.e.

$$\begin{pmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \end{pmatrix} = \begin{pmatrix} -\frac{k}{I} & \frac{Lmg}{I} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \end{pmatrix} + \begin{pmatrix} \tilde{\tau} \\ 0 \end{pmatrix},$$

General ODE case

Characterise the system dynamics by an arbitrary function f

$$\frac{dx(t)}{dt} = f(x(t)) \tag{5}$$

Expand f into a Taylor series (may not always be a good approximation)

$$f(x) = f(x_0) + \frac{df(x_0)}{dx}(x - x_0) + \frac{1}{2}\frac{d^2 f(x_0)}{dx^2}(x - x_0)^2 + \frac{1}{6}\frac{d^3 f(x_0)}{dx^3}(x - x_0)^3 + \dots$$

Often we can consider linear terms only (and constant terms) as a linearisation of the form (5)

$$\frac{dx(t)}{dt} = f(x_0) + \frac{df(x_0)}{dx}(x - x_0)$$

This is also how we get the dynamical matrix, i.e. $b_{ij} = \frac{df_i(x_0)}{dx_j}$, for the multidimensional linear case.

What if $\frac{df(x_0)}{dx} \approx 0$ (much smaller than higher-order terms)?

ODE (Open Dynamics Engine) http://ode.org

Means also Ordinary Differential Equation

- Written in C++ (internal), C (user interface) by Russel Smith (2000s)
- Physics engine (graphics engine is not even part of ODE): Rigid body dynamics simulation engine with collision detection (plus friction, grouping, constraints, joints etc.)
- Elements: box, sphere, capsule, mesh, cylinder
- Perfect for robotics (Run "webots" on DICE)
- Stability-accuracy trade-off?
- By now largely outdated

Collision handling in ODE

- Choice of convex pieces is useful for this purpose
- Collision detection before each simulation step gives a list of contact points
- Determine surface normal and distance (can be negative).
- A special contact joint is created for each contact point.
- The contact joints are put in a joint "group".
- A simulation step is taken.
- All contact joints are removed from the system.
- Collision functions are add-on and can be replaced by other collision detection libraries.

More about this later

Conclusion

- Continuous time appears to mean also continuous behaviour: *Natura non facit saltus* [Leibniz, Darwin]. In other words: Processes (such as collisions) need to be considered at an appropriate time-scale.
- Subtle difference exists between discrete and continuous case.
- To include control forces, we need to consider 2nd order systems.
- Linearisation is a way to deal with non-linear dynamics at a slow time scale (frequent updates).

Next topics

- Numerical integration
- Non-linearity and chaos
- Stochastic systems
- Iterated functions
- Trouble shooting