# 01622 Advanced Dynamical Systems: Applications in Science and Engineering

## Week 4: Time delays

Tobias K. S. Ritschel,
Asst. Prof. in Stochastic Adaptive Control

Department of Applied Mathematics and Computer Science,
Technical University of Denmark

Last updated on February 6, 2025

# Delay differential equations

# Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{1}$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad i = 1, \ldots, m \tag{2}$$

# How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0,1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \,\mathrm{d}s = 1 + t \tag{4}$$

Solution for $t \in [1,2]$

$$x(t) = x(1) + \int_1^t \overbrace{x(s-1)}^{=1+(s-1)} \,\mathrm{d}s = 2 + \frac{1}{2}(t^2 - 1^2) = \frac{3}{2} + \frac{1}{2}t^2 \tag{5}$$

Derivatives

$$\dot{x}(t) = \begin{cases} 0, & t \leq 0, \\ 1, & t \in [0,1], \\ t, & t \in [1,2], \end{cases} \qquad \ddot{x}(t) = \begin{cases} 0, & t \leq 0, \\ 0, & t \in [0,1], \\ 1, & t \in [1,2] \end{cases} \tag{6}$$

# Steady states

In steady state, $x(t) = x_s$ for all $t$

Steady state equations

$$0 = f(x_s, x_s, \ldots, x_s, u_s, d_s, p) \tag{7}$$

The steady state is the same as for ordinary differential equations in the form

$$\dot{x}(t) = f(x(t), x(t), \ldots, x(t), u(t), d(t), p) \tag{8}$$

Conclusion: Time delays do not change the steady state

# Stability – Linear systems

For linear systems, e.g., in the form

$$\dot{x}(t) = A(p)x(t) + G(p)x(t-\tau) + B(p)u(t) + E(p)d(t) \quad (9)$$

the stability is determined by $A$, $G$, and the time delay $\tau$

Characteristic equation

$$P(\lambda) = \det\left(A + Ge^{-\tau\lambda} - \lambda I\right) = 0 \quad (10)$$

In general, infinitely many solutions

# Graphical stability analysis

Real and imaginary parts of characteristic function

$$P_r(\lambda) = \operatorname{Re} P(\lambda), \qquad P_i(\lambda) = \operatorname{Im} P(\lambda) \tag{11}$$

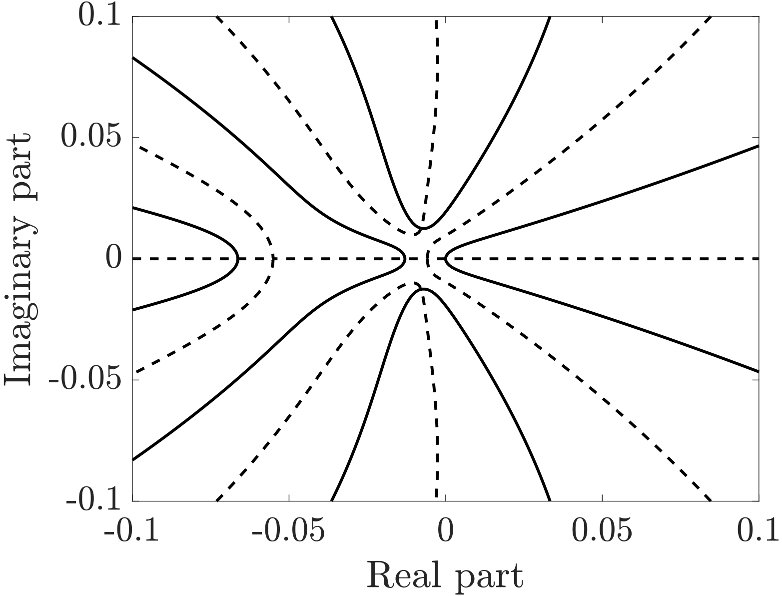
Choose a grid of complex values

$$\lambda_{mn} = a_m + ib_n, \qquad m = 1, \ldots, M, \qquad n = 1, \ldots, N \tag{12}$$

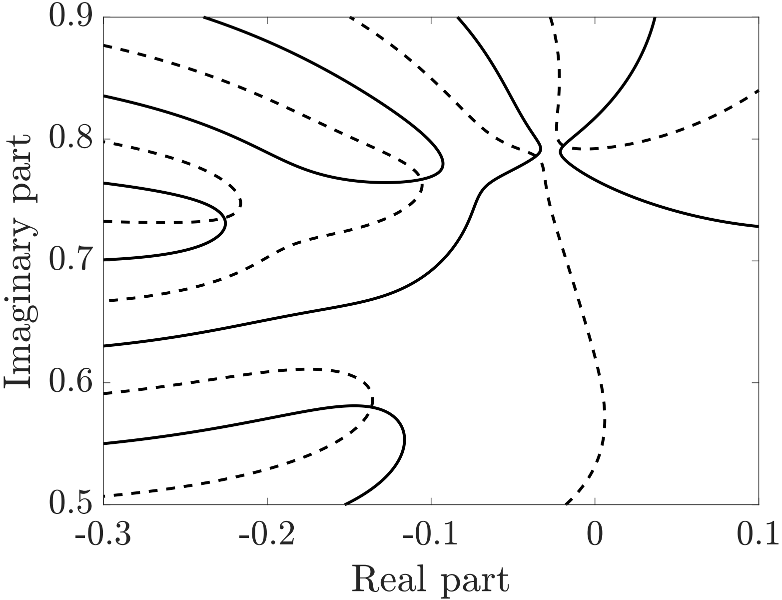
Plot the zero-contours of $P_r$ and $P_i$

The intersection between the contours indicate the roots

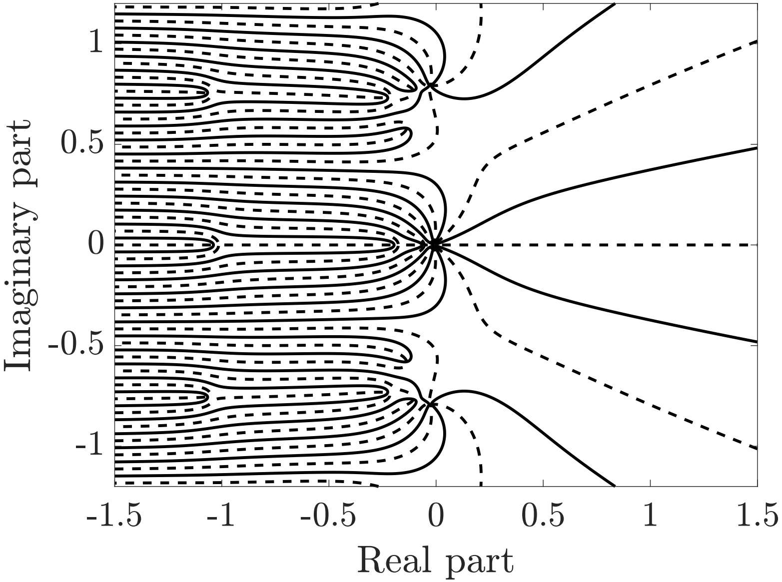
# Graphical stability analysis



# Graphical stability analysis



# Graphical stability analysis



# Numerical computation of the roots

Decision variables: $a$ and $b$

Algebraic equations

$$F(a,b) = P_r(a+ib) = 0, \tag{13a}$$
$$G(a,b) = P_i(a+ib) = 0 \tag{13b}$$

Two nonlinear equations in two variables that can be solved using, e.g., Matlab's fsolve

Initial guess: Use the graphical analysis

# Stability – Nonlinear systems

For nonlinear systems in the general form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{14}$$

the stability is determined by $A$, $G_i$, and $\tau_i$ for $i = 1, \ldots, m$

Characteristic equation

$$P(\lambda) = \det\left(A + \sum_{i=1}^{m} G_i e^{-\tau_i \lambda} - \lambda I\right) = 0 \tag{15}$$

Matrices

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \ldots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad G_i = \frac{\partial f}{\partial z_i} = \begin{bmatrix} \frac{\partial f_1}{\partial z_{i,1}} & \ldots & \frac{\partial f_1}{\partial z_{i,k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z_{i,1}} & \ldots & \frac{\partial f_n}{\partial z_{i,k}} \end{bmatrix}, \tag{16}$$

$i = 1, \ldots, k$

# Linearization – Linear systems

Linear system

$$\dot{x}(t) = Ax(t) + Gx(t-\tau) + Bu(t) + Ed(t) \tag{17}$$

Linearize the delayed state

$$x(t-\tau) \approx x(t) + \dot{x}(t)(t-\tau-t) = x(t) - \tau\dot{x}(t) \tag{18}$$

Approximate linear system

$$\dot{x}(t) = Ax(t) + G(x(t) - \tau\dot{x}(t)) + Bu(t) + Ed(t), \tag{19a}$$

$$(I+\tau G)\dot{x}(t) = (A+G)x(t) + Bu(t) + Ed(t) \tag{19b}$$

If $I + \tau G$ is invertible

$$\dot{x}(t) = \bar{A}x(t) + \bar{B}u(t) + \bar{E}d(t), \quad \bar{A} = (I+\tau G)^{-1}(A+G), \tag{20}$$

$$\bar{B} = (I+\tau G)^{-1}B, \quad \bar{E} = (I+\tau G)^{-1}E \tag{21}$$

# Linearization – Differential-algebraic equations

If $I + \tau G$ is not invertible

$$M\dot{x}(t) = \hat{A}x(t) + Bu(t) + Ed(t), \quad M = I + \tau G, \quad \hat{A} = A + G \tag{22}$$

Characteristic equation (generalized eigenvalue problem)

$$P(\lambda) = \det(\hat{A} - \lambda M) = 0 \tag{23}$$

Can be solved using, e.g., Matlab's eig

```
lambda = eig(Ahat, M)
```

# Linearization – Comparison with original stability criterion

Characteristic function

$$P(\lambda) = \det(\hat{A} - \lambda M) = \det(A + G - \lambda(I + \tau G))$$
$$= \det(A + G(1 - \tau\lambda) - \lambda I) \quad (24)$$

Original characteristic function

$$P(\lambda) = \det\left(A + Ge^{-\tau\lambda} - \lambda I\right) = 0 \quad (25)$$

Linearization of the exponential function (assume $x$ is small)

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{1}{2}x^2 + \cdots \approx 1 + x \quad (26)$$

Linearization of the exponential function in the original characteristic function

$$e^{-\tau\lambda} \approx 1 - \tau\lambda \quad (27)$$

# Linearization – Nonlinear systems

Nonlinear system

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{28}$$

Linearize the memory states

$$z_i(t) = x(t - \tau_i) \approx x(t) - \tau_i \dot{x}(t) \tag{29}$$

Jacobian matrices

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad G_i = \frac{\partial f}{\partial z_i} = \begin{bmatrix} \frac{\partial f_1}{\partial z_{i,1}} & \cdots & \frac{\partial f_1}{\partial z_{i,k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z_{i,1}} & \cdots & \frac{\partial f_n}{\partial z_{i,k}} \end{bmatrix}$$

Characteristic equation

$$\begin{aligned} P(\lambda) &= \det \left( A + \sum_{i=1}^{m} G_i - \left( I + \sum_{i=1}^{m} G_i \tau_i \right) \lambda \right) \\ &= \det \left( A + \sum_{i=1}^{m} G_i (1 - \tau_i \lambda) - \lambda I \right) = 0 \end{aligned} \tag{30}$$

# Nuclear reactor models

# Nuclear reactor model 6 – Model 4 revisited

Reactivity

$$\dot{\rho}(t) = -\kappa H C_n(t) \tag{31}$$

Mass balance equations

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{32a}$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} C_n(t) - \lambda_i C_i(t) + (C_{i,in}(t) - C_i(t))D \tag{32b}$$

Inlet concentration

$$C_{i,in}(t) = e^{-\lambda_i \tau} C_i(t - \tau) \tag{33}$$

Dilution rate

$$D = \frac{F}{V}, \qquad F = Av, \qquad \tau = L/v \tag{34}$$

# Nuclear reactor model 7 – Model 5 revisited

Reactivity

$$\dot{\rho}(t) = -\kappa \dot{T}_r(t) \tag{35}$$

Mass balance equations

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{36a}$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} C_n(t) - \lambda_i C_i(t) + (C_{i,in}(t) - C_i(t)) D \tag{36b}$$

Energy balance equations

$$\dot{T}_r(t) = \frac{f(t)}{n_r}(T_{hx}(t - \tau/2) - T_r(t)) + \frac{Q_g(t)}{n_r c_P}, \tag{37a}$$

$$\dot{T}_{hx}(t) = \frac{f(t)}{n_{hx}}(T_r(t - \tau/2) - T_{hx}(t)) - \frac{k_{hx}}{n_{hx} c_P}(T_{hx}(t) - T_c) \tag{37b}$$

Time-varying time delays

# Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{38}$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad\qquad i = 1, \ldots, m \tag{39}$$

Time-varying time delays

$$\tau_i = \tau_i(t), \qquad \tau_i = \tau_i(u(t)), \tag{40a}$$

$$\tau_i = \tau_i(x(t)), \qquad \tau_i = \tau_i(t, x(t), u(t), d(t), p) \tag{40b}$$

# What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

Thought experiment

1. Imagine two reactors that are connected by a pipe
2. Picture a model of the "receiving" reactor with time delay, $\tau$
   - The time delay is equal to length divided by velocity, $\tau = L/v$
3. Imagine that you reduce the velocity by a factor of 10
4. What is the true "age" of the content in the pipe?
5. What is the age of the inlet stream in the receiving reactor?

# Alternative to time delays: Transport equation

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{41}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t,s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t,s), \quad z_i(t,0) = x(t), \quad s \in [0,1], \tag{42}$$

Method of lines (first-order upwinded finite difference scheme)

$$z_{i,0}(t) = x(t), \tag{43a}$$

$$\dot{z}_{i,n}(t) = -\frac{1}{\tau_i}\frac{z_{i,n}(t) - z_{i,n-1}(t)}{\Delta s}, \quad \Delta s = \frac{1}{N}, \quad n = 1, \ldots, N, \tag{43b}$$

$$z_i(t) = z_{i,N}(t) \tag{43c}$$

The differential equations (41) and (43b) are ordinary

See [1] for more details and other ways to approximate time delays

# Open-loop simulation

# Numerical simulation

| Programming language | Simulator | Note |
|---|---|---|
| Matlab | dde23 | Constant time delays |
| Matlab | **ddesd** | General time delays |
| Matlab | ddensd | Neutral DDEs |
| Python | **JiTCDDE**[1] | General time delays |

[1] `https://jitcdde.readthedocs.io/en`

# Open-loop simulation

System

$$\dot{x}(t) = f(x(t), z_1(t), \dots, z_m(t), u(t), d(t), p), \tag{44a}$$
$$z_i(t) = x(t - \tau_i), \quad i = 1, \dots, m \tag{44b}$$

Zero-order hold parametrization

$$u(t) = u_k, \qquad t \in [t_k, t_{k+1}[, \tag{45a}$$
$$d(t) = d_k, \qquad t \in [t_k, t_{k+1}[ \tag{45b}$$

Open-loop simulation:

1. Create a function that, for given time $t$, returns $u_k$ and $d_k$, and call dde23/ddesd/JiTCDDE once for all control intervals
2. For each control interval, use the solution structure from the previous call to dde23/ddesd/JiTCDDE as the "history" input

Questions?

# Bibliography I


[1] T. K. S. Ritschel, A. T. Reenberg, P. E. Carstensen, J. Bendsen, and J. B. Jørgensen, "Mathematical meal models for simulation of human metabolism." arXiv: 2307.16444, 2024. Preprint.

[2] J. Hale, *Theory of functional differential equations*, vol. 3. Springer, 2nd ed., 1977.

[3] V. Kolmanovskii and A. Myshkis, *Applied theory of functional differential equations*, vol. 85 of *Mathematics and Its Applications*. Springer, 1992.

[4] S.-I. Niculescu and K. Gu, eds., *Advances in time-delay systems*, vol. 38 of *Lecture Notes in Computational Science and Engineering*. Springer, 2004.

[5] A. Bellen and M. Zennaro, *Numerical methods for delay differential equations*. Numerical Mathematics and Scientific Computation, Oxford University Press, 2003.

# Bibliography II


[6] N. H. Du, V. H. Linh, and V. Mehrmann, "Robust stability of differential-algebraic equations," in *Surveys in Differential-Algebraic Equations I* (A. Ilchmann and T. Reis, eds.), Differential-Algebraic Equations Forum, pp. 63–95, Springer, 2013.

[7] T. K. S. Ritschel and S. Stange, "Numerical optimal control for delay differential equations: A simultaneous approach based on linearization of the delayed state." arXiv:2410.02687, 2024. Preprint.

[8] T. K. S. Ritschel, "Numerical optimal control for distributed delay differential equations: A simultaneous approach based on linearization of the delayed variables." arXiv:2410.15083, 2024. Preprint.

[9] T. K. S. Ritschel and J. Wyller, "An algorithm for distributed time delay identification based on a mixed Erlang kernel approximation and the linear chain trick." arXiv:2405.07328, 2024. Preprint.