Linear dynamical systems

Purdue ME 597, Distributed Energy Resources

Kevin J. Kircher

Outline

Continuous-time linear dynamical systems

Linearization

Time discretization

Example: A simple climate model

A continuous-time linear dynamical system (LDS)

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = A(t)x(t) + B(t)u(t) + w(t)$$

- $t \in \mathbf{R}$ denotes time
- $x(t) \in \mathbf{R}^{n_x}$ is the **state**
- $u(t) \in \mathbf{R}^{n_u}$ is the **action** or **control**
- $w(t) \in \mathbf{R}^{n_x}$ is the **disturbance**
- $A(t) \in \mathbf{R}^{n_x \times n_x}$ is the **dynamics matrix**
- $B(t) \in \mathbf{R}^{n_x \times n_u}$ is the **action matrix** or **control matrix**

A continuous-time LDS with imperfect observations

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = A(t)x(t) + B(t)u(t) + w(t)$$
$$y(t) = C(t)x(t) + D(t)u(t) + v(t)$$

- $y(t) \in \mathbf{R}^{n_y}$ is the **observation** or **output**
- $v(t) \in \mathbf{R}^{n_y}$ is the **noise**
- $C(t) \in \mathbf{R}^{n_y \times n_x}$ is the **observation matrix**
- $D(t) \in \mathbf{R}^{n_y \times n_u}$ is the **feedthrough matrix**

Common simplifications

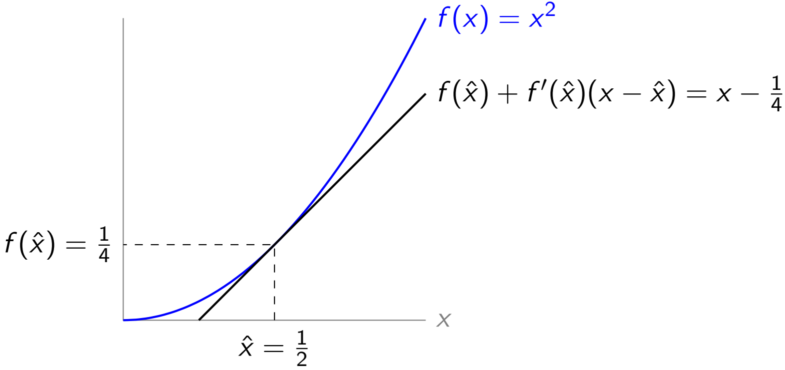
- **time-invariant**: A, B, C, and D are independent of t
- **single-input, single-output**: $n_u = n_y = 1$
- **no feedthrough**: $D(t) = 0$ for all t
- **perfectly observed**: $y(t) = x(t)$
- **deterministic**: $w(t) = 0$ and $v(t) = 0$ for all t

Outline

Reminder: Linearizing scalar-valued functions of scalars

- suppose nonlinear $f : \mathbf{R} \to \mathbf{R}$ is differentiable at $\hat{x} \in \mathbf{R}$
- Taylor's theorem: if x is near $\hat{x}$, then $f(x)$ is very near

$$f(\hat{x}) + f'(\hat{x})(x - \hat{x})$$



Linearizing vector-valued functions of vectors

- suppose nonlinear $f : \mathbf{R}^n \to \mathbf{R}^m$ is differentiable at $\hat{x} \in \mathbf{R}^n$
- Taylor's theorem: if x is near $\hat{x}$, then $f(x)$ is very near

$$f(\hat{x}) + D_f(\hat{x})(x - \hat{x})$$

where

$$D_f(\hat{x}) = \begin{bmatrix} \left.\frac{\partial f_1}{\partial x_1}\right|_{\hat{x}} & \cdots & \left.\frac{\partial f_1}{\partial x_n}\right|_{\hat{x}} \\ \vdots & & \vdots \\ \left.\frac{\partial f_m}{\partial x_1}\right|_{\hat{x}} & \cdots & \left.\frac{\partial f_m}{\partial x_n}\right|_{\hat{x}} \end{bmatrix} \in \mathbf{R}^{m \times n}$$

is the derivative (Jacobian) matrix of f at $\hat{x}$

Linearizing dynamical systems

- consider the nonlinear vector ODE

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u(t), w(t))$$

with dynamics function $f : \mathbf{R}^{n_x} \times \mathbf{R}^{n_u} \times \mathbf{R}^{n_w} \to \mathbf{R}^{n_x}$

- suppose at each t, $\hat{x}(t)$, $\hat{u}(t)$, and $\hat{w}(t)$ satisfy

$$\frac{\mathrm{d}\hat{x}(t)}{\mathrm{d}t} = f(\hat{x}(t), \hat{u}(t), \hat{w}(t))$$

(we call $\hat{x}$, $\hat{u}$, and $\hat{w}$ **nominal trajectories**)

- define the perturbations

$$\delta_x(t) = x(t) - \hat{x}(t),\ \delta_u(t) = u(t) - \hat{u}(t),\ \delta_w(t) = w(t) - \hat{w}(t)$$

Linearizing dynamical systems (continued)

- if $(x(t), u(t), w(t)) \approx (\hat{x}(t), \hat{u}(t), \hat{w}(t))$, then

$$\begin{aligned}
\frac{\mathrm{d}\delta_x(t)}{\mathrm{d}t} &= \frac{\mathrm{d}x(t)}{\mathrm{d}t} - \frac{\mathrm{d}\hat{x}(t)}{\mathrm{d}t} \\
&= f(x(t), u(t), w(t)) - f(\hat{x}(t), \hat{u}(t), \hat{w}(t)) \\
&\approx A(t)\delta_x(t) + B(t)\delta_u(t) + G(t)\delta_w(t)
\end{aligned}$$

  where

$$\begin{aligned}
A_{ij}(t) &= \left.\frac{\partial f_i}{\partial x_j}\right|_{\hat{x}(t),\hat{u}(t),\hat{w}(t)} \\
B_{ij}(t) &= \left.\frac{\partial f_i}{\partial u_j}\right|_{\hat{x}(t),\hat{u}(t),\hat{w}(t)} \\
G_{ij}(t) &= \left.\frac{\partial f_i}{\partial w_j}\right|_{\hat{x}(t),\hat{u}(t),\hat{w}(t)}
\end{aligned}$$

- this is an LDS with state δ_x, action δ_u, and disturbance $G\delta_w$

Outline

Continuous-time linear dynamical systems

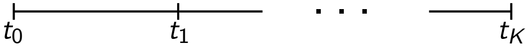
Linearization

Time discretization

Example: A simple climate model

Time discretization

- computers can simulate or optimize the evolution of LDS
- this is easiest if we divide the time span into discrete chunks



- K is the number of time steps
- $k \in \{0, \ldots, K\}$ indexes time steps
- often, we use a uniform time step Δt: $t_k = t_0 + k\Delta t$

Reminder: Solving first-order linear vector ODE IVPs

the solution to the first-order linear vector ODE IVP

$$x(t^{\mathrm{init}}) = x^{\mathrm{init}}, \;\; \frac{\mathrm{d}x(t)}{\mathrm{d}t} = Ax(t) + b(t)$$

with constant $A \in \mathbf{R}^{n \times n}$ is

$$x(t) = e^{(t - t^{\mathrm{init}})A} x^{\mathrm{init}} + e^{tA} \int_{t^{\mathrm{init}}}^{t} e^{-\tau A} b(\tau) \mathrm{d}\tau$$

Time discretization in general

- consider the perfectly observed LDS

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = A(t)x(t) + B(t)u(t) + w(t)$$

- suppose A is piecewise constant:

$$t_k \le t < t_{k+1} \implies A(t) = A(t_k)$$

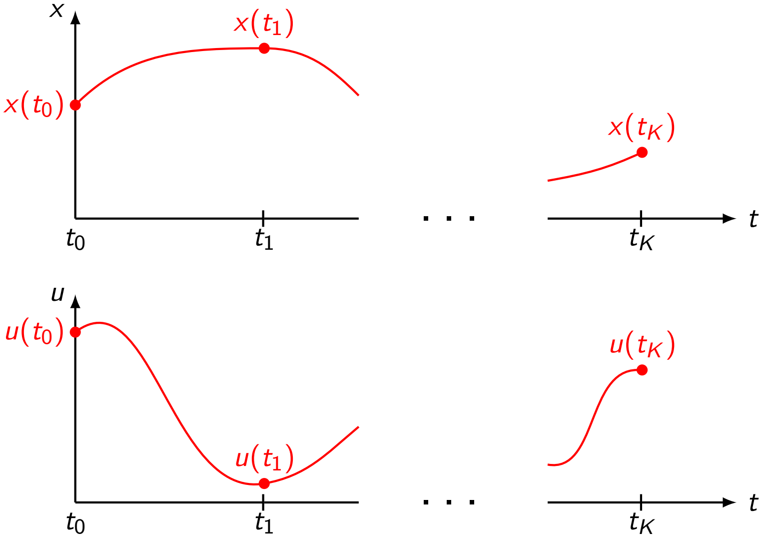
- then

$$\begin{aligned} x(t_{k+1}) = {} & e^{(t_{k+1}-t_k)A(t_k)}x(t_k) \\ & + e^{t_{k+1}A(t_k)} \int_{t_k}^{t_{k+1}} e^{-\tau A(t_k)}(B(\tau)u(\tau) + w(\tau))\mathrm{d}\tau \end{aligned}$$

- this is just the ODE IVP solution with $t^{\text{init}} = t_k$, $t = t_{k+1}$, and

$$b(t) = B(t)u(t) + w(t)$$

Time discretization in general



Time discretization with piecewise constant inputs

- if A, B, u, and w are piecewise constant,

$$t_k \le t < t_{k+1} \implies \begin{cases} A(t) = A(t_k),\ B(t) = B(t_k) \\ u(t) = u(t_k),\ w(t) = w(t_k), \end{cases}$$

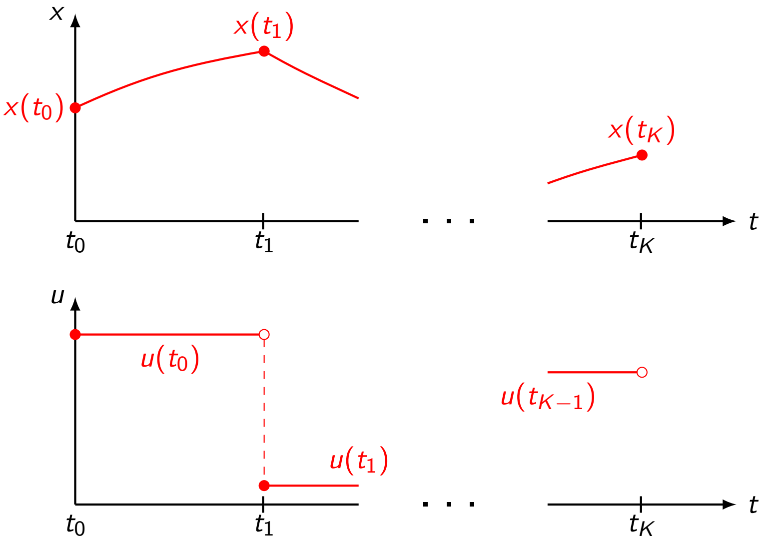
  then

$$\begin{aligned} x(t_{k+1}) &= e^{(t_{k+1}-t_k)A(t_k)}x(t_k) \\ &\qquad + e^{t_{k+1}A(t_k)} \int_{t_k}^{t_{k+1}} e^{-\tau A(t_k)} \mathrm{d}\tau (B(t_k)u(t_k) + w(t_k)) \end{aligned}$$

- if $A(t_k)$ is invertible, then

$$e^{t_{k+1}A(t_k)} \int_{t_k}^{t_{k+1}} e^{-\tau A(t_k)} \mathrm{d}\tau = \left(e^{(t_{k+1}-t_k)A(t_k)} - I \right) A(t_k)^{-1}$$

Time discretization with piecewise constant inputs



Summary: Discretizing LDS

- consider the continuous-time LDS

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \tilde{A}(t)x(t) + \tilde{B}(t)u(t) + \tilde{w}(t)$$

  with piecewise constant $\tilde{A}$, $\tilde{B}$, u, $\tilde{w}$

- the equivalent discrete-time LDS is

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k)$$

  where $\cdot(k)$ denotes $\cdot(t_k)$, $A(k) = e^{(t_{k+1}-t_k)\tilde{A}(t_k)}$, and

$$B(k) = e^{t_{k+1}\tilde{A}(t_k)} \int_{t_k}^{t_{k+1}} e^{-\tau\tilde{A}(t_k)}\mathrm{d}\tau \tilde{B}(t_k)$$

$$w(k) = e^{t_{k+1}\tilde{A}(t_k)} \int_{t_k}^{t_{k+1}} e^{-\tau\tilde{A}(t_k)}\mathrm{d}\tau \tilde{w}(t_k)$$

Summary: Discretizing LDS (continued)

- sample Matlab discretization code:

```
csys = ss(Atk,Btk,Ctk,Dtk); % continuous-time system
dsys = c2d(csys,t(k+1)-t(k)); % discrete-time system
Ak = dsys.A; % discrete-time dynamics matrix
```

- if the dynamics matrix $\tilde{A}(t_k)$ is invertible, then

$$B(k) = (A(k) - I)\,\tilde{A}(t_k)^{-1}\tilde{B}(t_k)$$
$$w(k) = (A(k) - I)\,\tilde{A}(t_k)^{-1}\tilde{w}(t_k)$$

Discretizing nonlinear dynamical systems

- there is no general analytical formula for discretizing

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u(t), w(t))$$

  with an arbitrary nonlinear dynamics function f
- but numerical ODE solvers can do the trick
- Runge-Kutta 4th order method works well for most problems
- Matlab example with $f(x(t), u(t), w(t)) = x(t)u(t)^{w(t)} \in \mathbf{R}$:

```
fk = @(tk,xk) xk*u(k)^w(k); % dynamics function
[~,soln] = ode45(fk,[t(k),t(k+1)],x(k)); % solver call
x(k+1) = soln(end); % solution
```

Outline

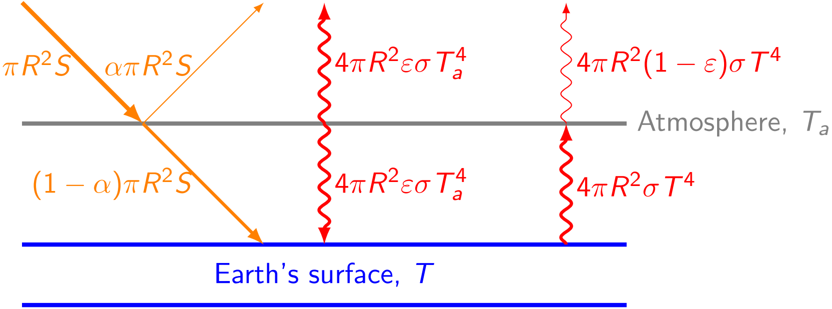
Continuous-time linear dynamical systems

Linearization

Time discretization

Example: A simple climate model

A simple model of earth's temperature dynamics



- orange is shortwave radiation (sunlight), red is longwave
- $R = 6.38 \times 10^6$ m is the earth's radius
- $S = 1370$ W/m^2 is the solar constant
- $\alpha = 0.3$, $\varepsilon = 0.767$ are the atmosphere's albedo, emissivity
- $\sigma = 5.67 \times 10^{-8}$ W/m^2/K^4 is the Stefan-Boltzmann constant

Assumptions

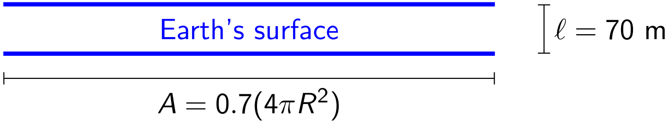
- "atmosphere" is very thin with negligible thermal capacitance

$\Longrightarrow$ its temperature responds instantly to changes in forcing

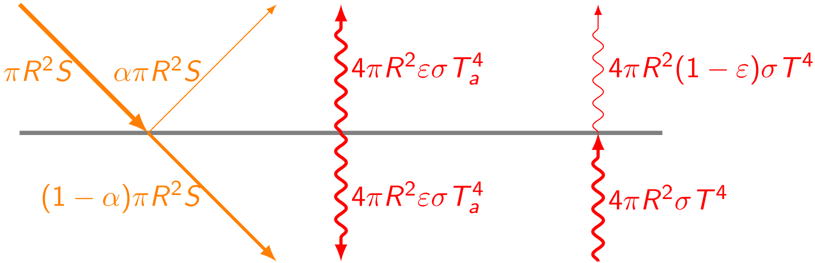
- "earth's surface" is 70 m of water covering 70% of surface

$\Longrightarrow$ its internal energy is $U = CT$ with thermal capacitance

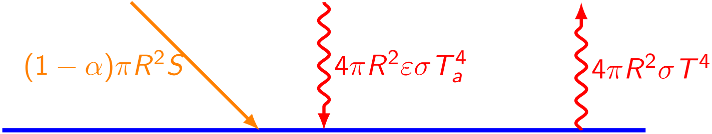
$$C = mc = \rho V c = \rho A \ell c = 1.05 \times 10^{23} \text{ J/K}$$



Steady-state power balance on atmosphere



$$\begin{aligned} &\text{power in} = \text{power out} \\ \iff \ & \pi R^2 (S + 4\sigma T^4) = \pi R^2 \left[\alpha S + (1-\alpha) S \right. \\ &\qquad \left. + 8\varepsilon\sigma T_a^4 + 4(1-\varepsilon)\sigma T^4\right] \\ \iff \ & T_a^4 = T^4/2 \end{aligned}$$



rate of change of energy = power in − power out

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \pi R^2 \left[(1-\alpha)S + 4\sigma\varepsilon T_a^4 - 4\sigma T^4)\right]$$

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{\pi R^2}{C} \left[(1-\alpha)S - 4\sigma(1-\varepsilon/2)T^4\right]$$

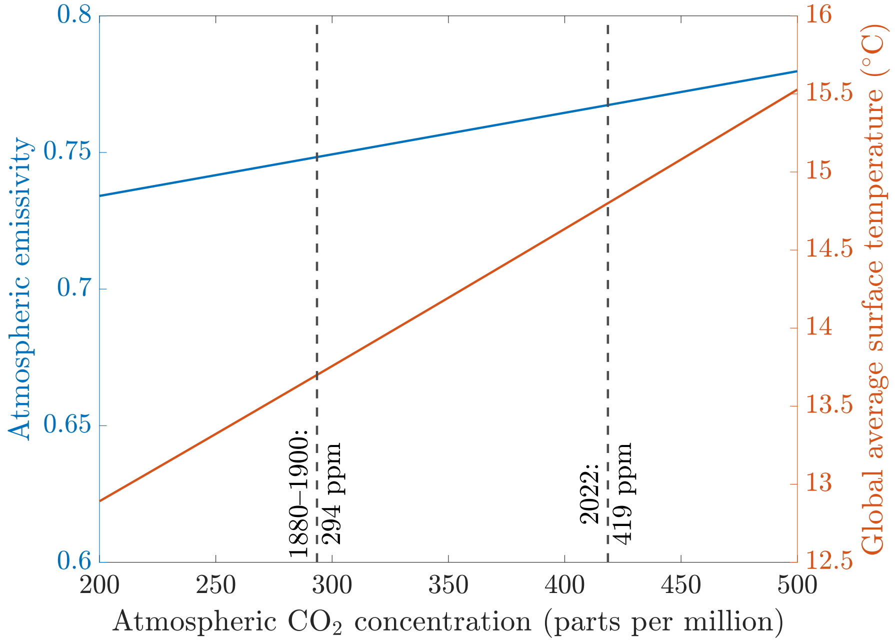
Effect of greenhouse gases on surface temperatures

- greenhouse gas emissions increase atmospheric emissivity ε
- in steady state, global-average surface temperature is

$$T = \sqrt[4]{\frac{(1-\alpha)S}{4\sigma(1-\varepsilon/2)}}$$

- if $\varepsilon = 0$, then $T = 255$ K $= -18$ °C $= -0.4$ °F
- if $\varepsilon = 1$, then $T = 303.3$ K $= 30.3$ °C $= 86.5$ °F
- 1880–1900 average: $T = 286.7$ K $= 13.7$ °C $= 56.7$ °F (consistent with an atmospheric emissivity of $\varepsilon = 0.748$)
- in 2022, T was 287.8 K $= 14.8$ °C $= 58.6$ °F (consistent with an atmospheric emissivity of $\varepsilon = 0.767$)

NOAA (2023), *Climate Change: Global Temperature*



Atmospheric emissivity
0.8
0.75
0.7
0.65
0.6
Global average surface temperature (°C)
16
15.5
15
14.5
14
13.5
13
12.5
200
250
300
350
400
450
500
Atmospheric CO_2 concentration (parts per million)
1880–1900:
294 ppm
2022:
419 ppm

Nonlinear dynamical system

dynamics:

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t} = \frac{\pi R^2}{C}\left[(1-\alpha(t))S - 4\sigma(1-\varepsilon(t)/2)T(t)^4\right]$$
$$\iff \frac{\mathrm{d}x(t)}{\mathrm{d}t} = \underbrace{-\beta(1-u(t)/2)x(t)^4 + \tilde{w}(t)}_{f(x(t),u(t),\tilde{w}(t))}$$

with

- state: $x(t) = T(t)$
- action: $u(t) = \varepsilon(t)$ (a stand-in for CO_2 concentration)
- (continuous-time) disturbance: $\tilde{w}(t) = \pi R^2(1-\alpha(t))S/C$
- parameter $\beta = 4\sigma\pi R^2/C$

Linearization

- given nominal $\hat{u}(t)$, $\hat{\tilde{w}}(t)$, compute nominal $\hat{x}(t)$ with ODE45
- the partial derivatives

$$\frac{\partial f}{\partial x(t)} = -4\beta(1 - u(t)/2)x(t)^3$$
$$\frac{\partial f}{\partial u(t)} = \beta x(t)^4/2, \;\; \frac{\partial f}{\partial \tilde{w}(t)} = 1$$

give linearized continuous-time dynamics

$$\delta_x(t) = \tilde{a}(t)\delta_x(t) + \tilde{b}(t)\delta_u(t) + \delta_{\tilde{w}}(t)$$

with $\delta_\cdot(t) = \cdot(t) - \hat{\cdot}(t)$ and

$$\tilde{a}(t) = -4\beta(1 - \hat{u}(t)/2)\hat{x}(t)^3, \;\; \tilde{b}(t) = \beta\hat{x}(t)^4/2$$

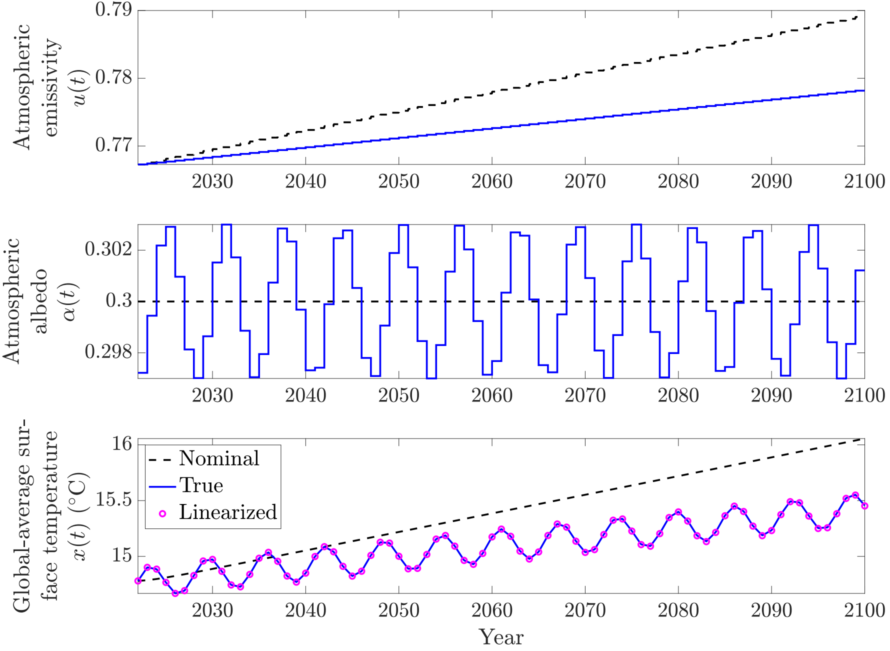
Time discretization

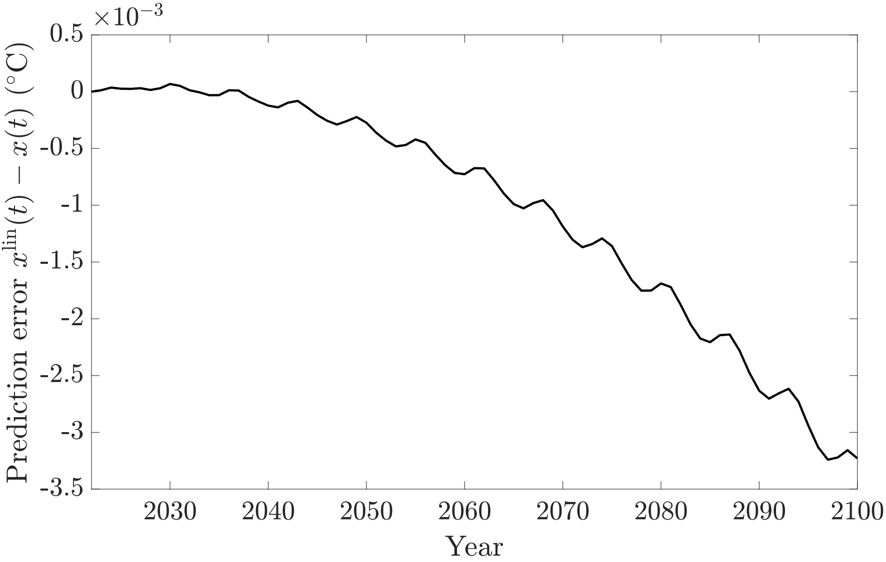
- use uniform time step Δt
- assume $\tilde{a}(t)$, $\tilde{b}(t)$, $\delta_u(t)$, $\delta_{\tilde{w}}(t)$ are piecewise constant
- then the discrete-time linearized system is

$$\delta_x(k+1) = a(k)\delta_x(k) + b(k)\delta_u(k) + \delta_w(k)$$

with

$$a(k) = e^{\Delta t \tilde{a}(t_k)}, \;\; b(k) = (a(k)-1)\,\tilde{b}(t_k)/\tilde{a}(t_k)$$
$$\delta_w(k) = (a(k)-1)\,\delta_{\tilde{w}}(t_k)/\tilde{a}(t_k)$$


0.79
0.78
0.77
Atmospheric emissivity $u(t)$
2030
2040
2050
2060
2070
2080
2090
2100
0.302
0.3
0.298
Atmospheric albedo $\alpha(t)$
16
15.5
15
Global-average surface temperature $x(t)$ (°C)
Nominal
True
Linearized
Year



- x^{lin} stays within 0.0035 °C of true x
- x^{lin} gets farther from x as x gets farther from nominal $\hat{x}$