Tutorial 8: Linear Systems

CSC2541 Tutorial 8, Winter 2022

Jenny Bao

Mar 3, 2022

Outline

- Linear, time-invariant (LTI) systems
- Time & frequency domain
  - Example: first-order optimizers
- Examples: frequency-domain insights using graphical tools
  - When does the momentum optimizer underdamp or overdamp? (problem set 1)
  - How robust is the momentum optimizer to gradient noises?

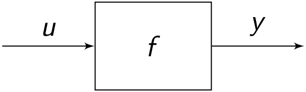
Block diagrams

The transformation from signal $u(t)$ to $y(t)$:

Example: ML optimizer

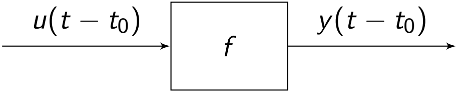
- $u(t)$: gradient at time t.
- $y(t)$: weight at time t.
- f: the optimization algorithm (e.g. gradient descent, heavy-ball momentum, ...)

Linear Time-Invariant system



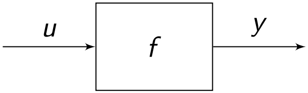
Linearity

Time-invariance



Why LTI systems?

"Linear systems are important because we can solve them."
—Richard Feynman

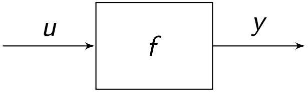


For an LTI system, let $f(t)$ be the impulse response of the system, we have:

$$y(t) = u(t) * f(t)$$

where $*$ denotes the convolution operation.

Representations of LTI systems



Time-domain:

- An intuitive representation for e.g. ML optimization
- Stability analysis often amounts to calculating eigenvalues

Frequency-domain:

- Classic theory developed in the 1930s
- Offers valuable insights through many graphical tools (e.g. root locus, Bode plot, Nyquist plot, ...)

Representations of LTI systems

$$y(t) = u(t) * f(t)$$

- Time domain (state-space)

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

- Frequency domain (input-output)
  - Convolution is very hard to compute directly!
  - Apply Laplace transform and turn it into multiplication

$$Y(s) = F(s)U(s)$$

Laplace transform

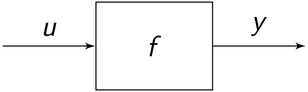
Laplace transform $\mathcal{L}\{\cdot\}$: $x(t) \to X(s)$.

$$X(s) = \mathcal{L}\{x(t)\} = \int_0^\infty x(t)e^{-st}dt$$

Important properties:

- Linearity: $a \cdot x(t) + b \cdot y(t) \to a \cdot X(s) + b \cdot Y(s)$
- Time-derivative: $\frac{d}{dt}x(t) \to sX(s)$
- Convolution: $x(t) * y(t) \to X(s)Y(s)$

Time domain → frequency-domain



Time-domain:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Frequency-domain (apply Laplace transform):

$$\begin{aligned} sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s) \end{aligned}$$

$$\implies Y(s) = \underbrace{\left(C(sI - A)^{-1}B + D \right)}_{F(s)} U(s)$$

$F(s)$ is the transfer function.

Discrete-time systems

- So far, we have only discussed continuous-time (CT) systems. However, most practical scenarios (including ML optimization) are discrete-time (DT).
- Fortunately, all of the above concepts have their discrete-time counterparts.

State-space representation (DT):

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned}$$

Frequency-domain with z-transform (DT counterpart of Laplace transform):

$$\begin{aligned} zX(z) &= AX(z) + BU(z) \\ Y(z) &= CX(z) + DU(z) \end{aligned}$$

$$\implies Y(z) = \underbrace{\left(C(zI - A)^{-1}B + D \right)}_{F(z)} U(z)$$

Example: gradient descent

$$w_{t+1} = w_t - \alpha g_t$$

- g_t: gradient at time t.
- w_t: weight at time t.

State-space (time domain) representation:

$$x_{t+1} = \underbrace{\begin{bmatrix} 1 \end{bmatrix}}_{A} x_t + \underbrace{\begin{bmatrix} -\alpha \end{bmatrix}}_{B} g_t$$

$$w_t = \underbrace{\begin{bmatrix} 1 \end{bmatrix}}_{C} x_t + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D} g_t$$

Example: gradient descent

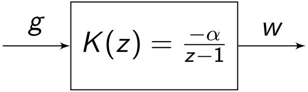
State-space (time domain) representation:

$$x_{t+1} = \underbrace{\begin{bmatrix}1\end{bmatrix}}_{A} x_t + \underbrace{\begin{bmatrix}-\alpha\end{bmatrix}}_{B} g_t$$

$$w_t = \underbrace{\begin{bmatrix}1\end{bmatrix}}_{C} x_t + \underbrace{\begin{bmatrix}0\end{bmatrix}}_{D} g_t$$

Frequency-domain representation (transfer function):

$$K(z) = C(zI - A)^{-1}B + D = \frac{-\alpha}{z-1}$$



In control theory, such a $K(z)$ is called an integral controller.

Example: gradient descent with momentum

$$w_{t+1} = w_t - \alpha g_t + \beta(w_t - w_{t-1})$$

Define $x_t = \begin{bmatrix} w_t \\ w_{t-1} \end{bmatrix}$, we have the state-space representation:

$$x_{t+1} = \underbrace{\begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix}}_{A} x_t + \underbrace{\begin{bmatrix} -\alpha \\ 0 \end{bmatrix}}_{B} g_t$$

$$w_t = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} x_t + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D} g_t$$

Example: gradient descent with momentum

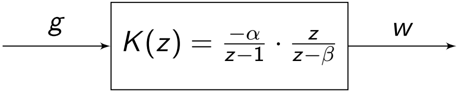
State-space (time domain) representation:

$$x_{t+1} = \underbrace{\begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix}}_{A} x_t + \underbrace{\begin{bmatrix} -\alpha \\ 0 \end{bmatrix}}_{B} g_t$$

$$w_t = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} x_t + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D} g_t$$

Frequency-domain representation (transfer function):

$$K(z) = C(zI - A)^{-1}B + D = \frac{-\alpha}{z-1} \cdot \frac{z}{z-\beta}$$

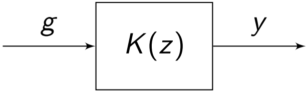


Exercise: Nesterov accelerated gradient

Express the Nesterov accelerated gradient as a state-space model and transfer function.

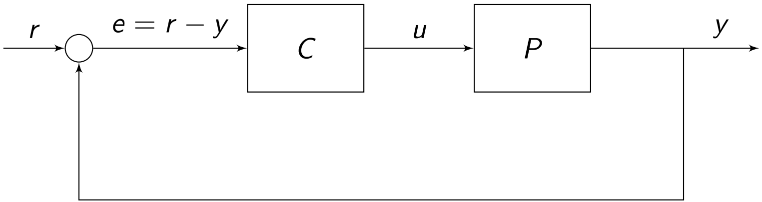
$$
\begin{aligned}
v_t &= \beta v_t - \alpha \nabla \mathcal{J}(w_t + \beta v_t) \\
w_{t+1} &= w_t + v_{t+1}
\end{aligned}
$$

Hint: the output of the system would be $y_t := w_t + \beta v_t$



Aside: PID control

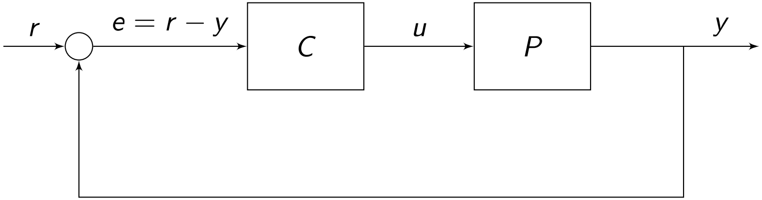
In control systems application, the single most celebrated controller is perhaps the **proportional–integral–derivative (PID)** controller. Consider the following block diagram:



Design controller C such that the error e is driven to 0 over time.

Aside: PID control

Design controller C such that the error e is driven to 0 over time.



PID control (continuous-time):

$$u(t) = \underbrace{K_p e(t)}_{proportional} + \underbrace{K_i \int_0^t e(\tau) d\tau}_{integral} + \underbrace{K_d \frac{de(t)}{dt}}_{derivative}$$

Optimizers & PID control

First-order optimizers are connected to PID control.

- Gradient descent

  $$\xrightarrow{g} \boxed{K(z) = \frac{-\alpha}{z-1}} \xrightarrow{w}$$

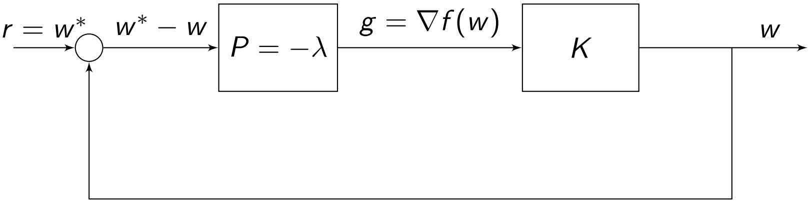
  This is integral control in discrete-time $(K_p, K_d = 0)$

- Heavy-ball momentum, Nesterov accelerated gradient can both be interpreted as variants of PID control. See `https://www.argmin.net/2018/04/19/pid/`.

Interconnected systems

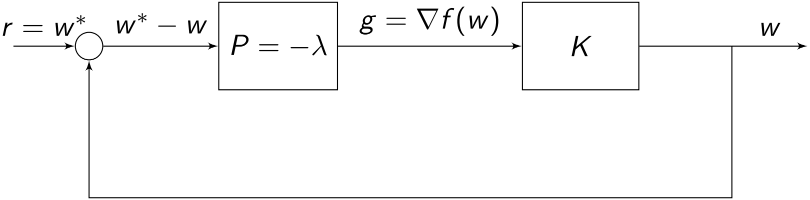
Expressing LTI systems in terms of transfer functions makes it convenient to study interconnected systems.

- For example, the optimization loop with quadratic objective function can be expressed in the following interconnected system:



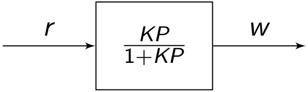
- P: transfer function from the weight to gradient. Since the objective is quadratic, we have $P = -\lambda$, where λ is th curvature.
- K the optimizer

Interconnected systems



We can compute the overall transfer function from r to w:

$$W(z) = K(z)P(z)(R(z) - W(z)) \quad \Longrightarrow \quad W(z) = \frac{KP}{1 + KP} R(z)$$



Frequency-domain analysis

Expressing the optimizers in frequency-domain provides many useful insights, such as:

- How does the optimizer perform in different curvature regimes of the objective function?
- How robust is the optimizer to gradient noise?

Many helpful graphical tools in the frequency-domain

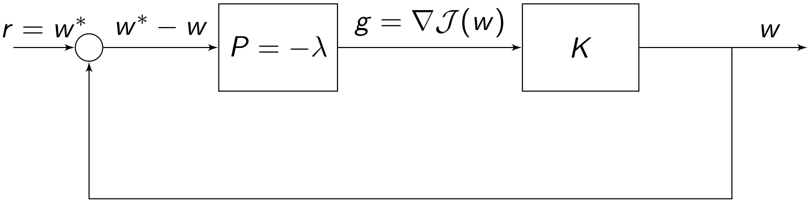
- Root locus, Bode plot, Nyquist plot, ...

Example: when does the momentum optimizer underdamp or overdamp?

The transfer function for heavy-ball momentum optimizer is:

$$K(z) = \frac{-\alpha}{z-1}\frac{z}{z-\beta}$$

- Let $\alpha = 0.01$, $\beta = 0.9$, and loss function be $\mathcal{J}(w) = \frac{1}{2}\lambda w^2$.



When is the interconnected system overdamped / underdamped / critically damped (as a function of curvature λ)?

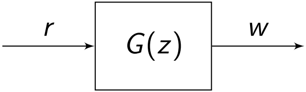
- You've done this in problem set 1.

Example: when does the momentum optimizer underdamp or overdamp?

Let's approach this using frequency-domain analysis.

- We know that the transfer function of the interconnected system is

$$G(z) = \frac{KP}{1+KP} = \frac{-\lambda K(z)}{1-\lambda K(z)}, \quad \text{where } K(z) = \frac{-\alpha}{z-1}\frac{z}{z-\beta}$$

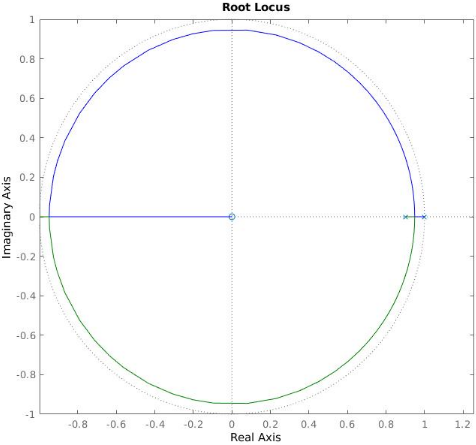


Fact: the roots of the denominator (a.k.a. poles) of $G(z)$ corresponds to the eigenvalues of matrix A in the state-space representation of G (i.e. the eigenvalues in problem set 1).

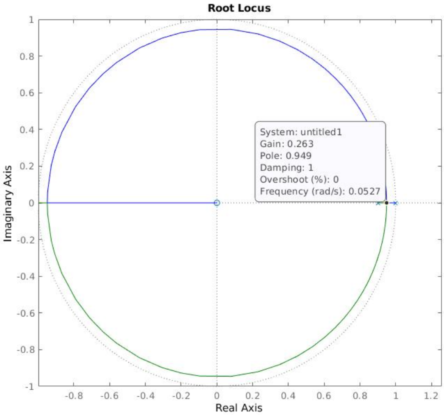
- We are interested in knowing how the poles of G(z) change as a function of curvature λ.
- A graphical tool called the root locus can help us with that.

Example: when does the momentum optimizer underdamp or overdamp?

Root locus plot showing how the poles evolve as functions of λ on the complex plane.

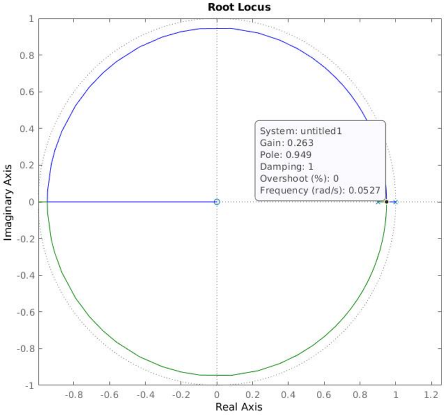


Example: when does the momentum optimizer underdamp or overdamp?



The plot shows that at $\lambda \approx 0.263$, the poles start to have imaginary parts.

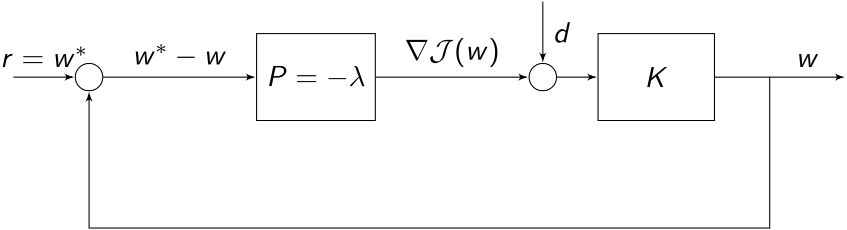
Example: when does the momentum optimizer underdamp or overdamp?



Check with the solution in problem set 1: the threshold is

$$T = \alpha^{-1}(1 - \sqrt{\beta})^2 \approx 0.263 \quad \checkmark$$

Example: how robust is the momentum optimizer to gradient noises?



How does the system respond to disturbance d (gradient noise) of different frequencies?

Example: how robust is the momentum optimizer to gradient noises?

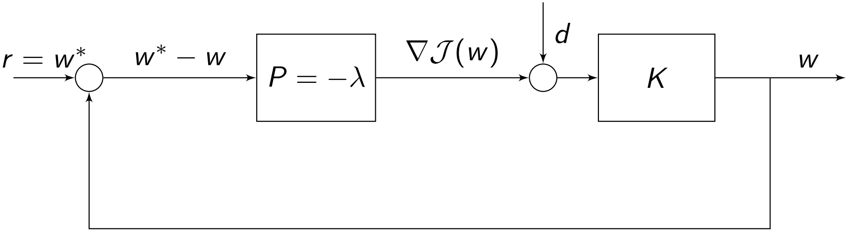
Property of LTI system:

- Sinusoid inputs are mapped to sinusoid outputs of the same frequency.

$$\sin(\omega t) \xrightarrow{\text{LTI}} a \sin(\omega t + \phi).$$

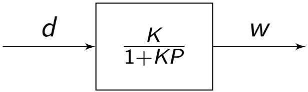
Bode plot: plots the magnitude a and phase ϕ as functions of ω.

Example: how robust is the momentum optimizer to gradient noises?



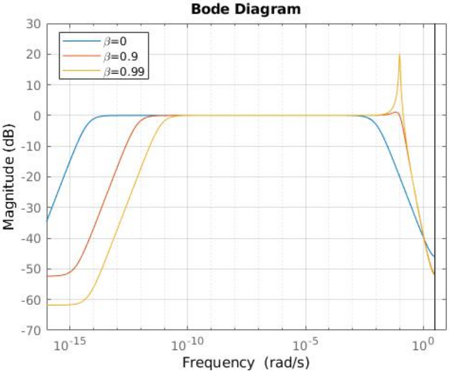
Write the transfer function from d to w (assume $r = 0$):

$$W = K(D + P(-W)) \implies W(z) = \frac{K}{1 + KP} D(z)$$



Example: how robust is the momentum optimizer to gradient noises?

Bode magnitude plot of $\frac{K}{1+KP}$, where $K(z) = \frac{-\alpha}{z-1}\frac{z}{z-\beta}$ and $P = -\lambda$ (let $\alpha = 0.01, \lambda = 1$).



As β increases, gradient noise at frequency around $0.1\mathrm{rad/s}$ is amplified more and more!

Summary

- LTI system basics
  - Block diagram, linearity & time-invariance
  - Time-domain & frequency-domain representations
    - Laplace transform, transfer functions
    - Example: gradient descent, heavy-ball momentum
    - Aside: optimizers & PID control
- Examples: frequency-domain analysis using graphical tools:
  - Underdamping / overdamping for momentum optimizer (root locus plot).
  - Robustness to gradient noises (Bode plot).