NA 565 - Fall 2023

# Linear Quadratic Control

September 13, 2023

UNIVERSITY OF MICHIGAN

$f(x) = \frac{1}{2}\|Ax - b\|^2$

- Gradient: $\nabla f(x) = A^\mathsf{T} Ax - A^\mathsf{T} b$
- Hessian: $H(x) = A^\mathsf{T} A$

**Assumption**

- $A \in \mathbb{R}^{m \times n}$ *and* $b \in \mathbb{R}^m$
- $m \geq n \Leftrightarrow A$ *is a tall matrix*
- $\mathrm{rank}(A) = n$ *(i.e., columns of $A$ are linearly independent)*

$A \in \mathbb{R}^{m \times n}$ has linearly independent columns $\Leftrightarrow A^\mathsf{T} A \succ 0$.

$$\nabla f(x^\star) = 0 \Rightarrow x^\star = (A^\mathsf{T} A)^{-1} A^\mathsf{T} b$$

- $A$ is full (column) rank $\Rightarrow A^\mathsf{T} A \succ 0$ is invertible
- Solve a linear system — "Normal Equations"

$$(A^\mathsf{T} A) x^\star = A^\mathsf{T} b$$

- Cholesky ($A^\mathsf{T} A = LL^\mathsf{T}$) or QR ($A = QR$) factorization

# Example: Linear Regression with $\ell_2$-Regularizer

Given a dataset $\{(x_i,t_i)\}_{i=1}^{N}$, where $x$ is the input and $t$ is the target (output), we wish to find a linear model that explains data. The model is linear in weights with nonlinear basis functions.

$$y(x;w) = \sum_{j=0}^{N} w_j \phi_j(x) = w^\mathsf{T} \phi(x),$$

$$w = \mathrm{vec}(w_0,w_1,\dots,w_N) \quad \text{and} \quad \phi = \mathrm{vec}(\phi_0,\phi_1,\dots,\phi_N),$$

$\phi_0 = 1$ and $w_0$ is a bias parameter. A common basis function is the Gaussian (Squared Exponential) basis

$$\phi_j(x) = \exp\left(-\frac{(x-x_j)^2}{2s^2}\right),$$

The hyperparameter $s$ is called the basis bandwidth or length-scale.

To find $w \in \mathbb{R}^{N+1}$, we solve the following regularized least squares problem.

$$\underset{w \in \mathbb{R}^{N+1}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{N} \Big( t_i - w^{\mathsf{T}} \phi(x_i) \Big)^2 + \frac{\lambda}{2} \|w\|^2,$$

or

$$\underset{w \in \mathbb{R}^{N+1}}{\text{minimize}} \quad f(w) := \frac{1}{2} \|t - \Phi w\|^2 + \frac{\lambda}{2} \|w\|^2,$$

where $t = \mathrm{vec}(t_1, \ldots, t_N)$ and $\Phi$ is a $N \times N + 1$ design matrix

$$\Phi = \begin{bmatrix} \phi^{\mathsf{T}}(x_1) \\ \vdots \\ \phi^{\mathsf{T}}(x_N) \end{bmatrix}.$$

# Example: Linear Regression with $\ell_2$-Regularizer

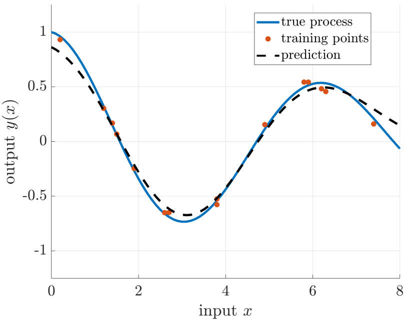
$$f(w) = \frac{1}{2}\|t - \Phi w\|^2 + \frac{\lambda}{2}\|w\|^2$$

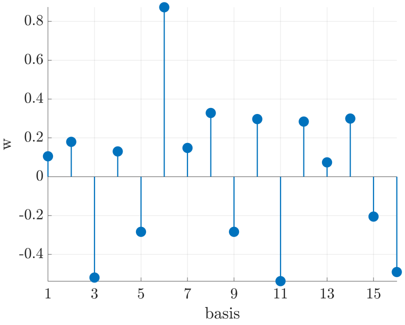
$$\nabla f(w) = \Phi^{\mathsf{T}}\Phi w - \Phi^{\mathsf{T}} t + \lambda w$$

$$\nabla f(w^\star) = 0 \Rightarrow \boxed{w^\star = (\Phi^{\mathsf{T}}\Phi + \lambda I)^{-1}\Phi^{\mathsf{T}} t}$$

# Example: Linear Regression with $\ell_2$-Regularizer

`https://github.com/UMich-CURLY-teaching/UMich-ROB-530-public/tree/main/code-examples/Python/linear_regression`



# Least squares with equality constraints

- the (linearly) *constrained least squares problem* (CLS) is

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|^2 \\ \text{subject to} & Cx = d \end{array}$$

- variable (to be chosen/found) is $n$-vector $x$
- $m \times n$ matrix $A$, $m$-vector $b$, $p \times n$ matrix $C$, and $p$-vector $d$ are *problem data* (*i.e.*, they are given)
- $\|Ax - b\|^2$ is the *objective function*
- $Cx = d$ are the *equality constraints*
- $x$ is *feasible* if $Cx = d$
- $\hat{x}$ is a *solution* of CLS if $C\hat{x} = d$ and $\|A\hat{x} - b\|^2 \leq \|Ax - b\|^2$ holds for any $n$-vector $x$ that satisfies $Cx = d$

# Least squares with equality constraints

- CLS combines solving linear equations with least squares problem
- like a bi-objective least squares problem, with infinite weight on second objective $\|Cx - d\|^2$

# Least norm problem

- special case of constrained least squares problem, with $A = I$, $b = 0$
- *least-norm problem*:

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & Cx = d \end{array}$$

  *i.e.*, find the smallest vector that satisfies a set of linear equations

## Optimality conditions via calculus

to solve constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) = \|Ax - b\|^2 \\ \text{subject to} & c_i^T x = d_i, \quad i = 1, \ldots, p \end{array}$$

1. form *Lagrangian* function, with *Lagrange multipliers* $z_1, \ldots, z_p$

$$L(x, z) = f(x) + z_1(c_1^T x - d_1) + \cdots + z_p(c_p^T x - d_p)$$

2. optimality conditions are

$$\frac{\partial L}{\partial x_i}(\hat{x}, z) = 0, \quad i = 1, \ldots, n, \qquad \frac{\partial L}{\partial z_i}(\hat{x}, z) = 0, \quad i = 1, \ldots, p$$

# Optimality conditions via calculus

- $\dfrac{\partial L}{\partial z_i}(\hat{x}, z) = c_i^T \hat{x} - d_i = 0$, which we already knew
- first $n$ equations are more interesting:

$$\frac{\partial L}{\partial x_i}(\hat{x}, z) = 2\sum_{j=1}^{n}(A^TA)_{ij}\hat{x}_j - 2(A^Tb)_i + \sum_{j=1}^{p} z_j c_i = 0$$

- in matrix-vector form: $2(A^TA)\hat{x} - 2A^Tb + C^Tz = 0$
- put together with $C\hat{x} = d$ to get *Karush–Kuhn–Tucker (KKT) conditions*

$$\left[\begin{array}{cc} 2A^TA & C^T \\ C & 0 \end{array}\right]\left[\begin{array}{c} \hat{x} \\ z \end{array}\right] = \left[\begin{array}{c} 2A^Tb \\ d \end{array}\right]$$

  a square set of $n + p$ linear equations in variables $\hat{x}$, $z$

- KKT equations are extension of normal equations to CLS

# Linear dynamical system

$$x_{t+1} = A_t x_t + B_t u_t, \quad y_t = C_t x_t, \quad t = 1, 2, \ldots$$

- $n$-vector $x_t$ is *state* at time $t$
- $m$-vector $u_t$ is *input* at time $t$
- $p$-vector $y_t$ is *output* at time $t$
- $n \times n$ matrix $A_t$ is *dynamics matrix*
- $n \times m$ matrix $B_t$ is *input matrix*
- $p \times n$ matrix $C_t$ is *output matrix*
- $x_t$, $u_t$, $y_t$ often represent deviations from a standard operating condition

# Linear quadratic control

$$\begin{array}{ll} \text{minimize} & J_{\text{output}} + \rho J_{\text{input}} \\ \text{subject to} & x_{t+1} = A_t x_t + B_t u_t, \quad t = 1, \ldots, T-1 \\ & x_1 = x^{\text{init}}, \quad x_T = x^{\text{des}} \end{array}$$

- variables are state sequence $x_1, \ldots, x_T$ and input sequence $u_1, \ldots, u_{T-1}$
- two objectives are quadratic functions of state and input sequences:

$$\begin{aligned} J_{\text{output}} &= \|y_1\|^2 + \cdots + \|y_T\|^2 = \|C_1 x_1\|^2 + \cdots + \|C_T x_T\|^2 \\ J_{\text{input}} &= \|u_1\|^2 + \cdots + \|u_{T-1}\|^2 \end{aligned}$$

- first constraint imposes the linear dynamics equations
- second set of constraints specifies the initial and final state
- $\rho$ is positive parameter used to trade off the two objectives

# Constrained least squares formulation

$$
\begin{array}{ll}
\text{minimize} & \|C_1 x_1\|^2 + \cdots + \|C_T x_T\|^2 + \rho\|u_1\|^2 + \cdots + \rho\|u_{T-1}\|^2 \\
\text{subject to} & x_{t+1} = A_t x_t + B_t u_t, \quad t = 1, \ldots, T-1 \\
& x_1 = x^{\text{init}}, \quad x_T = x^{\text{des}}
\end{array}
$$

- can be written as

$$
\begin{array}{ll}
\text{minimize} & \|\tilde{A}z - \tilde{b}\|^2 \\
\text{subject to} & \tilde{C}z = \tilde{d}
\end{array}
$$

- vector $z$ contains the $Tn + (T-1)m$ variables:

$$
z = (x_1, \ldots, x_T, u_1, \ldots, u_{T-1})
$$

# Constrained least squares formulation

$$\tilde{A} = \left[\begin{array}{ccc|ccc} C_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & C_T & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & \sqrt{\rho} I & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \sqrt{\rho} I \end{array}\right], \qquad \tilde{b} = 0$$

$$\tilde{C} = \left[\begin{array}{cccccc|cccc} A_1 & -I & 0 & \cdots & 0 & 0 & B_1 & 0 & \cdots & 0 \\ 0 & A_2 & -I & \cdots & 0 & 0 & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{T-1} & -I & 0 & 0 & \cdots & B_{T-1} \\ \hline I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & 0 & 0 & \cdots & 0 \end{array}\right], \qquad \tilde{d} = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline x^{\mathrm{init}} \\ x^{\mathrm{des}} \end{array}\right]$$

# Example

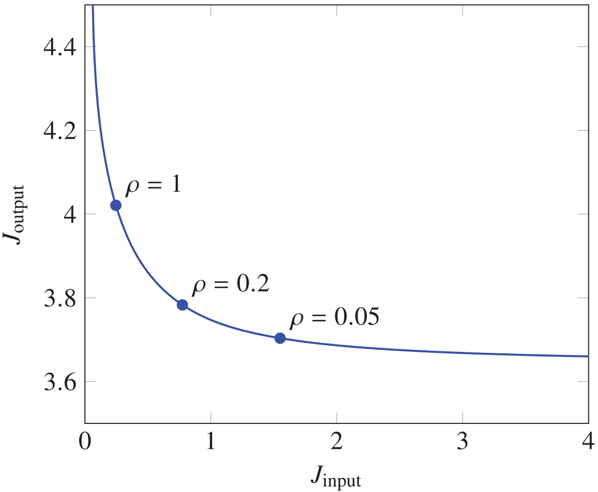
- time-invariant system: system matrices are constant

$$A = \begin{bmatrix} 0.855 & 1.161 & 0.667 \\ 0.015 & 1.073 & 0.053 \\ -0.084 & 0.059 & 1.022 \end{bmatrix}, \qquad B = \begin{bmatrix} -0.076 \\ -0.139 \\ 0.342 \end{bmatrix},$$

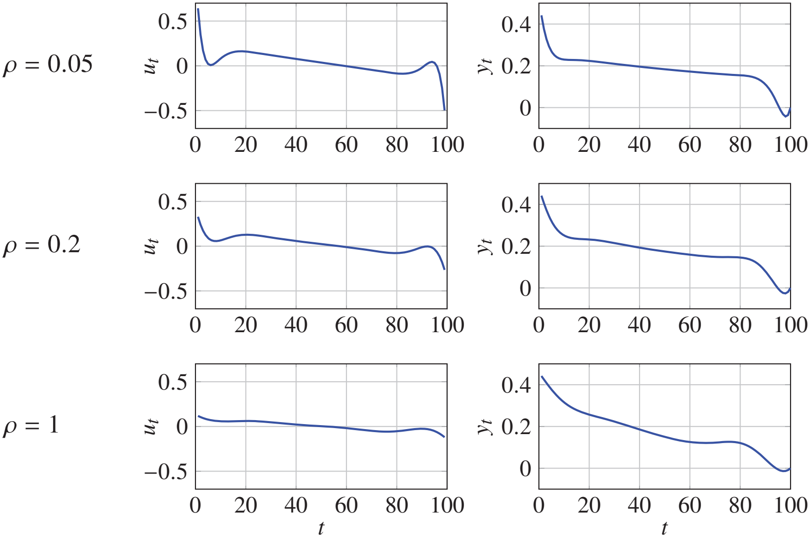
$$C = \begin{bmatrix} 0.218 & -3.597 & -1.683 \end{bmatrix}$$

- initial condition $x^{\text{init}} = (0.496, -0.745, 1.394)$
- target or desired final state $x^{\text{des}} = 0$
- $T = 100$

# Optimal trade-off curve



# Three points on the trade-off curve



## Linear state feedback control

- linear state feedback control uses the input

$$u_t = Kx_t, \quad t = 1, 2, \ldots$$

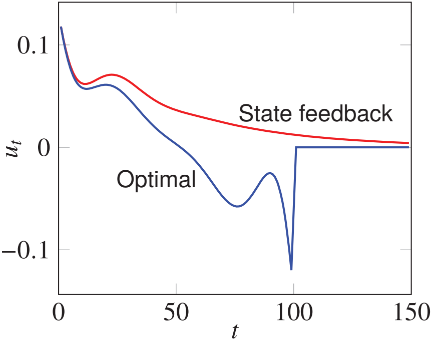
- $K$ is *state feedback gain matrix*
- widely used, especially when $x_t$ should converge to zero, $T$ is not specified

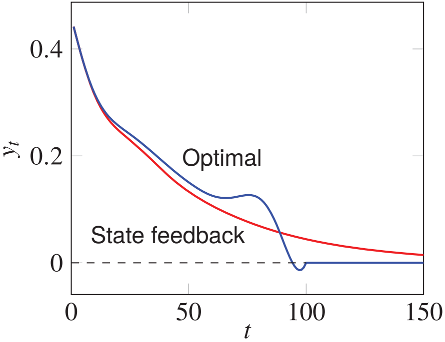
- one choice for $K$: solve linear quadratic control problem with $x^{\text{des}} = 0$
- solution $u_t$ is a linear function of $x^{\text{init}}$, so $u_1$ can be written as

$$u_1 = Kx^{\text{init}}$$

- columns of $K$ can be found by computing $u_1$ for $x^{\text{init}} = e_1, \ldots, e_n$
- use this $K$ as state feedback gain matrix

# Example




- system matrices of previous example
- blue curve uses optimal linear quadratic control for $T = 100$
- red curve uses simple linear state feedback $u_t = Kx_t$

# State estimation

- linear dynamical system model:

$$x_{t+1} = A_t x_t + B_t w_t, \quad y_t = C_t x_t + v_t, \quad t = 1, 2, \ldots$$

- $x_t$ is *state* ($n$-vector)
- $y_t$ is *measurement* ($p$-vector)
- $w_t$ is *input* or *process noise* ($m$-vector)
- $v_t$ is *measurement noise* or *measurement residual* ($p$-vector)
- we know $A_t$, $B_t$, $C_t$, and measurements $y_1, \ldots, y_T$
- $w_t, v_t$ are unknown, but assumed small
- *state estimation*: estimate/guess $x_1, \ldots, x_T$

# Least squares state estimation

$$\begin{array}{ll} \text{minimize} & J_{\text{meas}} + \lambda J_{\text{proc}} \\ \text{subject to} & x_{t+1} = A_t x_t + B_t w_t, \quad t = 1, \ldots, T-1 \end{array}$$

- variables: states $x_1, \ldots, x_T$ and input noise $w_1, \ldots, w_{T-1}$
- primary objective $J_{\text{meas}}$ is sum of squares of measurement residuals:

$$J_{\text{meas}} = \|C_1 x_1 - y_1\|^2 + \cdots + \|C_T x_T - y_T\|^2$$

- secondary objective $J_{\text{proc}}$ is sum of squares of process noise

$$J_{\text{proc}} = \|w_1\|^2 + \cdots + \|w_{T-1}\|^2$$

- $\lambda > 0$ is a parameter, trades off measurement and process errors

# Constrained least squares formulation

$$\begin{array}{ll} \text{minimize} & \|C_1x_1 - y_1\|^2 + \cdots + \|C_Tx_T - y_T\|^2 + \lambda(\|w_1\|^2 + \cdots + \|w_{T-1}\|^2) \\ \text{subject to} & x_{t+1} = A_tx_t + B_tw_t, \quad t = 1, \ldots, T-1 \end{array}$$

- can be written as

$$\begin{array}{ll} \text{minimize} & \|\tilde{A}z - \tilde{b}\|^2 \\ \text{subject to} & \tilde{C}z = \tilde{d} \end{array}$$

- vector $z$ contains the $Tn + (T-1)m$ variables:

$$z = (x_1, \ldots, x_T, w_1, \ldots, w_{T-1})$$

# Constrained least squares formulation

$$\tilde{A} = \left[\begin{array}{cccc|ccc} C_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & C_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & C_T & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & \sqrt{\lambda} I & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \sqrt{\lambda} I \end{array}\right], \qquad \tilde{b} = \left[\begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_T \\ \hline 0 \\ \vdots \\ 0 \end{array}\right]$$

$$\tilde{C} = \left[\begin{array}{cccccc|cccc} A_1 & -I & 0 & \cdots & 0 & 0 & B_1 & 0 & \cdots & 0 \\ 0 & A_2 & -I & \cdots & 0 & 0 & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{T-1} & -I & 0 & 0 & \cdots & B_{T-1} \end{array}\right], \qquad \tilde{d} = 0$$

# Missing measurements

- suppose we have measurements $y_t$ for $t \in \mathcal{T}$, a subset of $\{1, \ldots, T\}$
- measurements for $t \notin \mathcal{T}$ are missing
- to estimate states, use same formulation but with

$$J_{\text{meas}} = \sum_{t \in \mathcal{T}} \|C_t x_t - y_t\|^2$$

- from estimated states $\hat{x}_t$, can estimate missing measurements

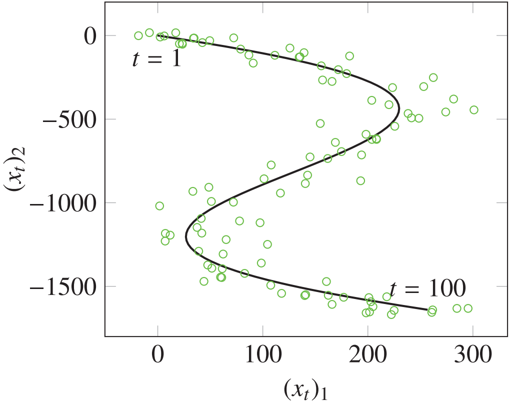
$$\hat{y}_t = C_t \hat{x}_t, \quad t \notin \mathcal{T}$$

# Example

$$A_t = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad B_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad C_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

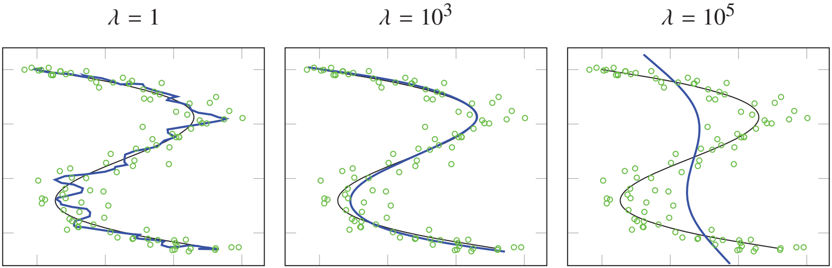
- simple model of mass moving in a 2-D plane
- $x_t = (p_t, z_t)$: $2$-vector $p_t$ is position, $2$-vector $z_t$ is the velocity
- $y_t = C_t x_t + w_t$ is noisy measurement of position
- $T = 100$

## Measurements and true positions




- solid line is exact position $C_t x_t$
- 100 noisy measurements $y_t$ shown as circles

# Position estimates

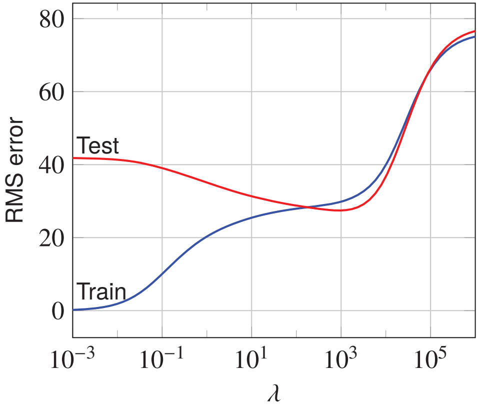



blue lines show position estimates for three values of $\lambda$

# Cross-validation

- randomly remove $20\%$ (say) of the measurements and use as test set
- for many values of $\lambda$, estimate states using other (*training*) measurements
- for each $\lambda$, evaluate RMS measurement residuals on test set
- choose $\lambda$ to (approximately) minimize the RMS test residuals

# Example




- cross-validation method applied to previous example
- remove 20 of the 100 measurements
- suggests using $\lambda \approx 10^3$

# Next Time

- Linear Quadratic Control Examples in Python