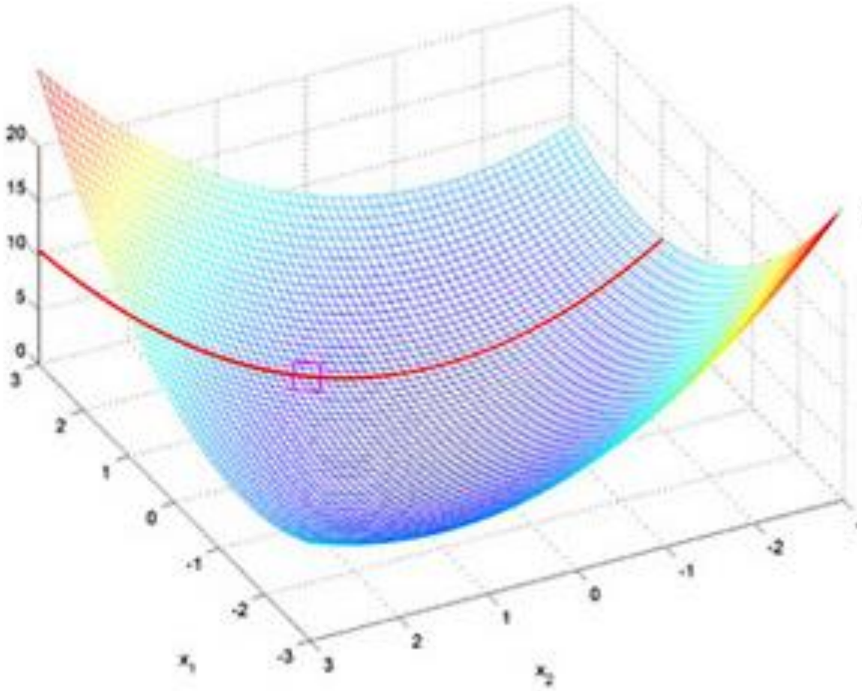
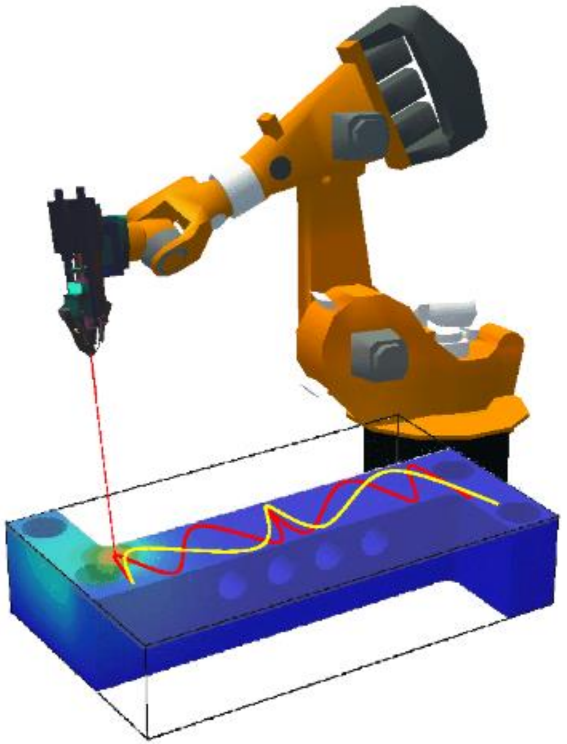
MEGR 3090/7090/8090: Advanced Optimal Control



$$\boxed{V_n(\mathbf{x}_n) = \min_{\{\mathbf{u}_n,\mathbf{u}_{n+1},\cdots,\mathbf{u}_{N-1}\}} \left[\frac{1}{2}\sum_{k=n}^{N-1}\left(\mathbf{x}_k^T\mathbf{Q}_k\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k\right) + \frac{1}{2}\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N\right]}$$

$$\begin{aligned}
V_n(\mathbf{x}_n) &= \min_{\{\mathbf{u}_n,\mathbf{u}_{n+1},\cdots,\mathbf{u}_{N-1}\}} \left[\frac{1}{2}\sum_{k=n}^{N-1}\left(\mathbf{x}_k^T\mathbf{Q}_k\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k\right) + \frac{1}{2}\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N\right] \\
&= \min_{\mathbf{u}_n}\left[\frac{1}{2}\left(\mathbf{x}_n^T\mathbf{Q}_n\mathbf{x}_n + \mathbf{u}_n^T\mathbf{R}\mathbf{u}_n\right) + \underbrace{\min_{\{\mathbf{u}_{n+1},\cdots,\mathbf{u}_{N-1}\}}\left[\frac{1}{2}\sum_{k=n+1}^{N-1}\left(\mathbf{x}_k^T\mathbf{Q}_k\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k\right) + \frac{1}{2}\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N\right]}_{V_{n+1}(\mathbf{x}_{n+1})}\right] \\
&= \min_{\mathbf{u}_n}\left[\frac{1}{2}\left(\mathbf{x}_n^T\mathbf{Q}_n\mathbf{x}_n + \mathbf{u}_n^T\mathbf{R}\mathbf{u}_n\right) + V_{n+1}(\mathbf{x}_{n+1})\right]
\end{aligned}$$

$$\boxed{V_n(\mathbf{x}_n) = \min_{\mathbf{u}_n}\left[\frac{1}{2}\left(\mathbf{x}_n^T\mathbf{Q}_n\mathbf{x}_n + \mathbf{u}_n^T\mathbf{R}\mathbf{u}_n\right) + V_{n+1}(\mathbf{x}_{n+1})\right]}$$



Lecture 5

September 5, 2017

Design Optimization vs. Optimal Control - Reminder

UNC CHARLOTTE

Design optimization framework:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} J(\mathbf{p})$$

where p = vector of design parameters and J is a **static function** of those design parameters

Subject to: $\mathbf{p} \in P$

Control optimization framework:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} J(\mathbf{u}; \mathbf{x}(0))$$

where

$$J\big(\mathbf{u}; \mathbf{x}(0)\big) = \sum_{i=0}^{N-1} g(\mathbf{x}(i), u(i)) + h(\mathbf{x}(N))$$

Subject to:

$$\mathbf{x}(i+1) = f(\mathbf{x}(i), \mathbf{u}(i))$$
$$\mathbf{u}(i) \in U, i = 0 \ldots N-1$$
$$\mathbf{x}(i) \in X, i = 0 \ldots N-1$$
$$\mathbf{x}(N) \in X_f$$

Key point: The Papalambros textbook ***only*** addresses design optimization – however, we have shown that the discrete-time, finite horizon (finite N) control optimization is ***equivalent*** to the design optimization problem

Roadmap for Finite-horizon, Discrete-time Optimal Control

Unconstrained optimization fundamentals (today):

- Existence/
- uniqueness of minima
- Convexity

Unconstrained convex optimization tools:

- Gradient descent
- Newton’s method

Constrained convex optimization fundamentals:

- Convex sets
- KKT conditions
- Lagrange multipliers

Constrained convex optimization tools:

- Linear and quadratic programming
- Sequential quadratic programming (SQP)

Non-convex optimization through dynamic programming:

- Bellman’s principle of optimality
- State and control quantization (“meshing”)

Papalambros Chapter 4

Papalambros Chapter 5

Papalambros Chapter 5 & 7

Kirk Chapter 3

Unconstrained Optimization - Setup

UNC CHARLOTTE

Design optimization framework:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} J(\mathbf{p})$$

where $\mathbf{p}$ = vector of design parameters and J is a **static function** of those design parameters

Subject to: $\mathbf{p} \in \mathbb{R}^{\dim(\mathbf{p})}$

Control optimization framework:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} J(\mathbf{u}; \mathbf{x}(0))$$

where

$$J\big(\mathbf{u}; \mathbf{x}(0)\big) = \sum_{i=0}^{N-1} g(\mathbf{x}(i), \mathbf{u}(i)) + h(\mathbf{x}(N))$$

Subject to:

$$\mathbf{u}(i) \in \mathbb{R}^{\dim(\mathbf{u})}, i = 0 \ldots N-1$$

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), u(k))$$

Key point: The parameter vector (in the case of design optimization) and control input vector at each step (in the case of control optimization) can be **any vector of real numbers**

Unconstrained Optimization - Setup

From this point forward, $\underline{u}$ = decision variable

in our notes, regardless of whether we are faced with a design optimization or optimal control problem.
Note: Papalambros likes to use $\mathbf{x}$ for the decision variable, but it will be clear from context.

Global vs. Local Optima

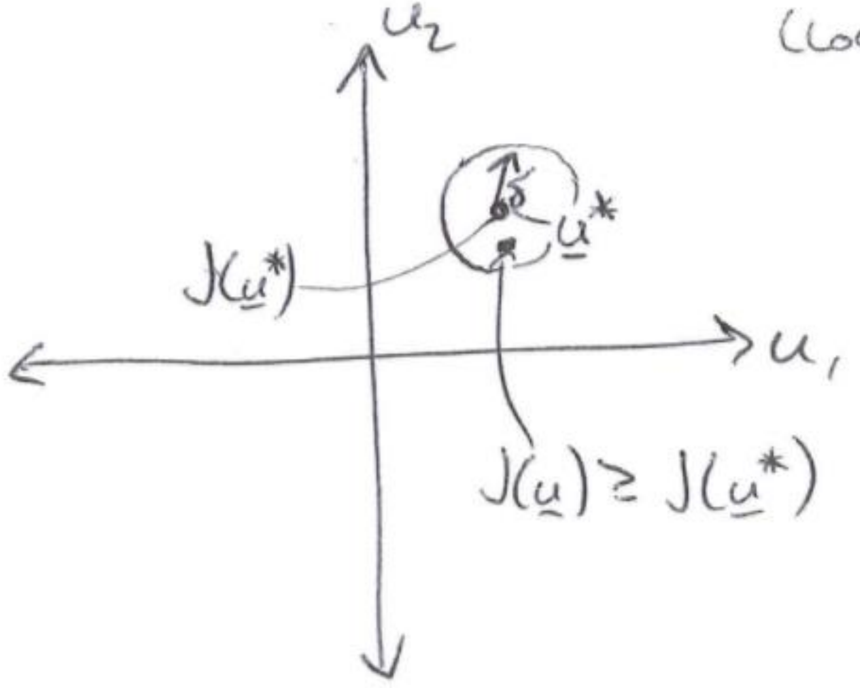
Consider a continuous function, $J(\mathbf{u})$, where $\mathbf{u} \in \mathbb{R}^p$ and $J(\mathbf{u}) \in \mathbb{R}$:

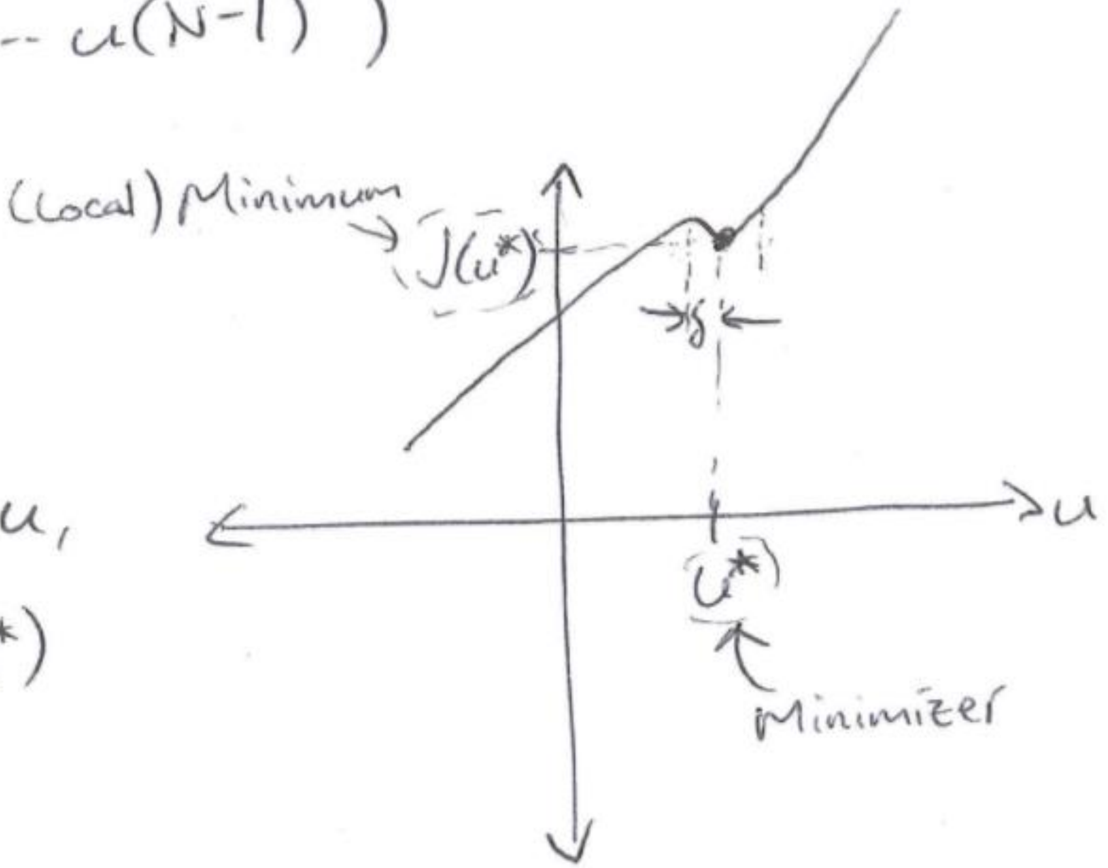
- $\mathbf{u}^*$ is said to be a **local optimum** (and $J(\mathbf{u}^*)$ is the corresponding **local minimum**) if and only if there exists a scalar $\delta > 0$ such that $J(\mathbf{u}) \geq J(\mathbf{u}^*)$ whenever $\|\mathbf{u} - \mathbf{u}^*\| \leq \delta$ and $\mathbf{u} \neq \mathbf{u}^*$. Note that $\mathbf{u}^*$ could also be referred to as a **local minimizer.**

- $\mathbf{u}^*$ is said to be a **global optimum** (and $J(\mathbf{u}^*)$ is the corresponding **global minimum**) if and only $J(\mathbf{u}) \geq J(\mathbf{u}^*)$ whenever $\mathbf{u} \neq \mathbf{u}^*$. Note that $\mathbf{u}^*$ could also be referred to as a **global minimizer.**

Global vs. Local Optima

$$\|\underline{u}\| = \left(u(0)^2 + u(1)^2 + \cdots u(N-1)^2\right)^{1/2}$$





First-Order Necessity Condition for Local and Global Optima

UNC CHARLOTTE

Suppose that a continuous function, $J(\mathbf{u})$, possesses a local minimum denoted by J^*, at $\mathbf{u}^*$ (i.e., $J^* \triangleq J(\mathbf{u}^*)$). Then it **must** be true that $\nabla J(\mathbf{u}^*) = \mathbf{0}$, where $\nabla J(\mathbf{u}^*) = \left[\frac{\partial J}{\partial u_1} \quad \cdots \quad \frac{\partial J}{\partial u_p}\right]_{\mathbf{u}^*}$

Key points:

- $\nabla J(\mathbf{u}^*)$ is a **row vector** whose dimension is equal to that of $\mathbf{u}$.
- When u is a scalar, $\nabla J(\mathbf{u}^*) = \left.\frac{dJ}{du}\right|_{u^*}$, leading to the familiar necessity condition from calculus 1: $\left.\frac{dJ}{du}\right|_{u^*} = 0$ at an optimum.
- For **unconstrained** optimizations, any **finite global** optimum is also a **local** optimum.

First-Order Necessity Condition – Simple Examples

UNC CHARLOTTE

For the systems below, determine the **possible optima** based on the first-order necessity condition:

Example 1: $J(u) = (u-7)^2$ … Hopefully a review from calc 1!

Example 2: $J(\mathbf{u}) = 2u_1^2 + 3u_2^2 + 3u_1u_2 + 4u_1$

First-Order Necessity Condition – Simple Examples

Ex. 1: $J(u) = (u-7)^2$

$$\nabla J = \frac{dJ}{du} = 2(u-7)$$

$$u^* - 7 = 0 \Rightarrow u^* = 7 \quad \underline{\text{maybe}}$$

Ex. 2: $J(\underline{u}) = 2u_1^2 + 3u_2^2 + 3u_1u_2 + 4u_1$

$$\nabla J = [4u_1 + 3u_2 + 4 \quad 6u_2 + 3u_1]$$

Need
$$\left.\begin{aligned} 4u_1^* + 3u_2^* + 4 &= 0 \\ 6u_2^* + 3u_1^* &= 0 \end{aligned}\right\} \underbrace{\begin{bmatrix} 4 & 3 \\ 3 & 6 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix}}_{\underline{u}^*} = \underbrace{\begin{bmatrix} -4 \\ 0 \end{bmatrix}}_{b}$$

$$A\underline{u}^* = b$$

$$\underline{u}^* = A^{-1}b$$

$$= [-1.6 \quad 0.8]^T \quad \underline{\text{maybe}}$$

First-Order Necessity Condition – Optimal Control Example

Consider the following discrete-time system model: $x(k+1) = x(k) + u(k)$

Objective: Minimize $J\big(\mathbf{u}; x(0)\big) = \sum_{i=0}^{2} [x(i)^2 + u(i)^2]$

Given: $x(0) = 10$

Tasks:

- Compute $\nabla J(u)$
- Given the first-order necessity condition, compute the **possible** value(s) of $\mathbf{u}^*$ (i.e., compute the possible optimal control trajectories)

First-Order Necessity Condition – Optimal Control Example

$$\text{Ex: } J(\underline{u}; x(0)) = \sum_{i=0}^{2} \left[x(i)^2 + u(i)^2 \right]$$

Given $x(0) = 10$ where $x(k+1) = x(k) + u(k)$

$$x(1) = x(0) + u(0)$$

$$x(2) = x(1) + u(1) = x(0) + u(0) + u(1)$$

$$\Rightarrow J(\underline{u}; x(0)) = x(0)^2 + u(0)^2 + [x(0) + u(0)]^2 + u(1)^2 + [x(0) + u(0) + u(1)]^2 + u(2)^2$$

$$= 100 + u(1)^2 + u(2)^2 + u(0)^2 + [10 + u(0)]^2 + [10 + u(0) + u(1)]^2$$

First-Order Necessity Condition – Optimal Control Example

$$\nabla J = \begin{bmatrix} 2u(0) + 2[10+u(0)] + 2[10+u(0)+u(1)] \\ 2u(1) + 2[10+u(0)+u(1)] \\ 2u(2) \end{bmatrix}^T$$

$$\Rightarrow \quad 2u^*(0) + 2[10+u^*(0)] + 2[10+u^*(0)+u^*(1)] = 0$$

$$2u^*(1) + 2[10+u^*(0)+u^*(1)] = 0$$

$$2u^*(2) = 0 \Rightarrow u^*(2) = 0 \checkmark$$

$$\Rightarrow \begin{bmatrix} 6 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} u^*(0) \\ u^*(1) \end{bmatrix} = \begin{bmatrix} -40 \\ -20 \end{bmatrix}$$

$$\Rightarrow \underline{u}^* = [-6 \quad -2 \quad 0]^T \quad \text{maybe}$$

Second-Order Sufficiency Conditions for Local and Global Optima

Suppose that $\nabla J(\mathbf{u}^*) = \mathbf{0}$. Then:

- $\mathbf{u}^*$ is a local optimum (i.e., a local minimizer) if $J(\mathbf{u})$ is **locally convex** around $\mathbf{u}^*$
- $\mathbf{u}^*$ is a global optimum (i.e., a global minimizer) if $J(\mathbf{u})$ is **globally convex**

Definitions of convexity (graphical interpretations to follow):

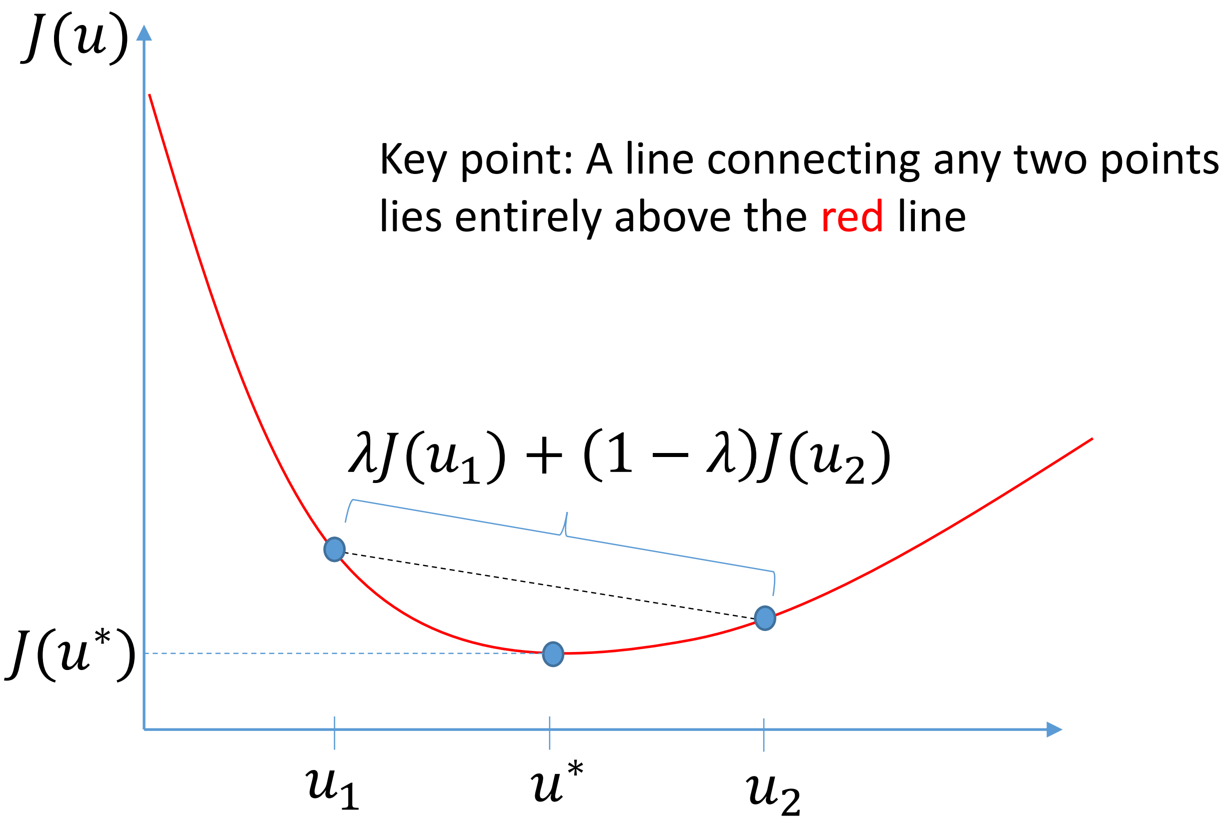
- A function $J(\mathbf{u})$ is **locally convex around $\mathbf{u}^*$** if there exists a scalar $\delta > 0$ such that $J(\lambda \mathbf{u_1} + (1-\lambda)\mathbf{u_2}) \leq \lambda J(\mathbf{u_1}) + (1-\lambda)J(\mathbf{u_2})$ for all $\mathbf{u_1}, \mathbf{u_2}, \lambda$ satisfying $\|\mathbf{u_1} - \mathbf{u}^*\| \leq \delta$, $\|\mathbf{u_2} - \mathbf{u}^*\| \leq \delta$, $0 < \lambda < 1$. $J(\mathbf{u})$ is said to be **locally *strictly* convex around $\mathbf{u}^*$** if $J(\lambda \mathbf{u_1} + (1-\lambda)\mathbf{u_2}) < \lambda J(\mathbf{u_1}) + (1-\lambda)J(\mathbf{u_2})$ for the same set of conditions on $\mathbf{u_1}, \mathbf{u_2}, \lambda$.

- A function $J(\mathbf{u})$ is **globally convex** if $J(\lambda \mathbf{u_1} + (1-\lambda)\mathbf{u_2}) \leq \lambda J(\mathbf{u_1}) + (1-\lambda)J(\mathbf{u_2})$ for all $\mathbf{u_1}, \mathbf{u_2}$, and for all λ satisfying $0 < \lambda < 1$. $J(\mathbf{u})$ is said to be **locally *strictly* convex** if $J(\lambda \mathbf{u_1} +$

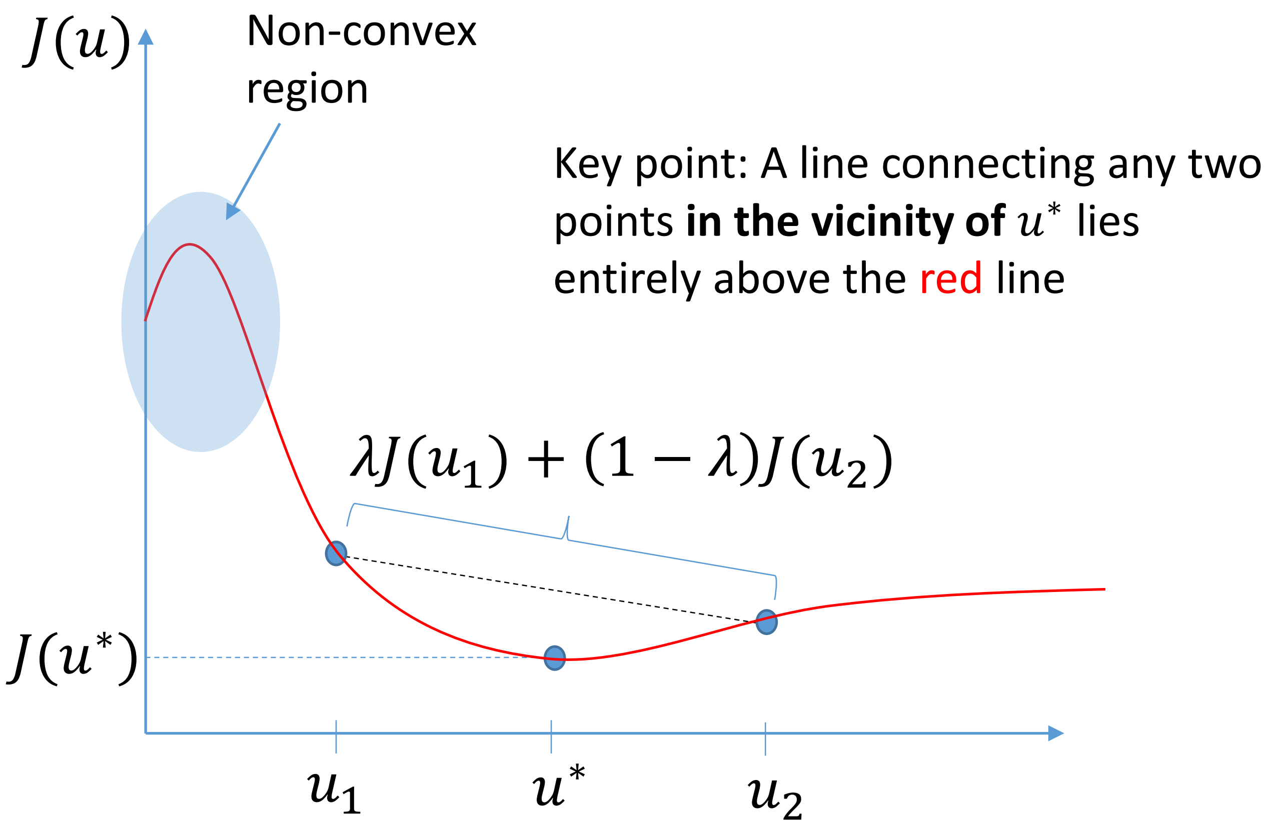
Convexity of Functions – Scalar Graphical Interpretation

Reminder: For convexity, we want $J(\lambda u_1 + (1-\lambda)u_2) \leq \lambda J(u_1) + (1-\lambda)J(u_2), 0 < \lambda < 1$

Example **globally** (as far as we can tell) convex function:



Example **locally** convex function around u^*:

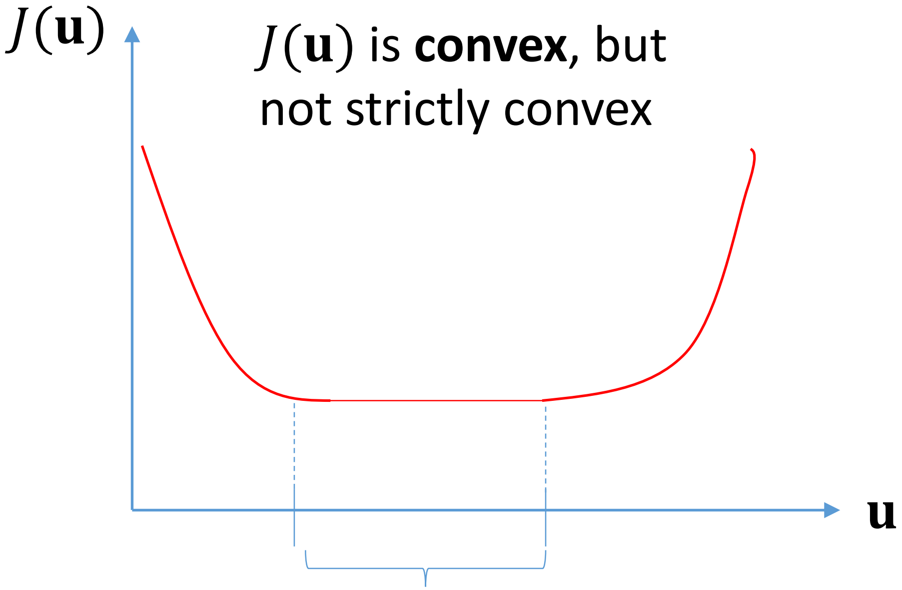


Second-Order Sufficiency Conditions for Local and Global Optima – Reminder and More Details

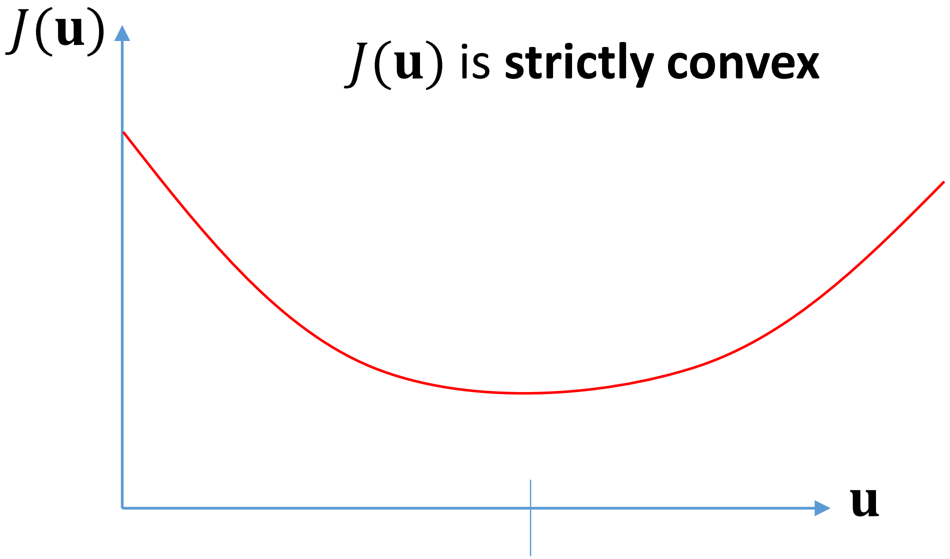
UNC CHARLOTTE

Suppose that $\nabla J(\mathbf{u}^*) = \mathbf{0}$. Then:

- $\mathbf{u}^*$ is a local optimum (i.e., a local minimizer) if $J(\mathbf{u})$ is **locally convex** around $\mathbf{u}^*$
- $\mathbf{u}^*$ is a global optimum (i.e., a global minimizer) if $J(\mathbf{u})$ is **globally convex**
- The above optima are **unique** if $J(\mathbf{u})$ is ***strictly* locally/globally convex**



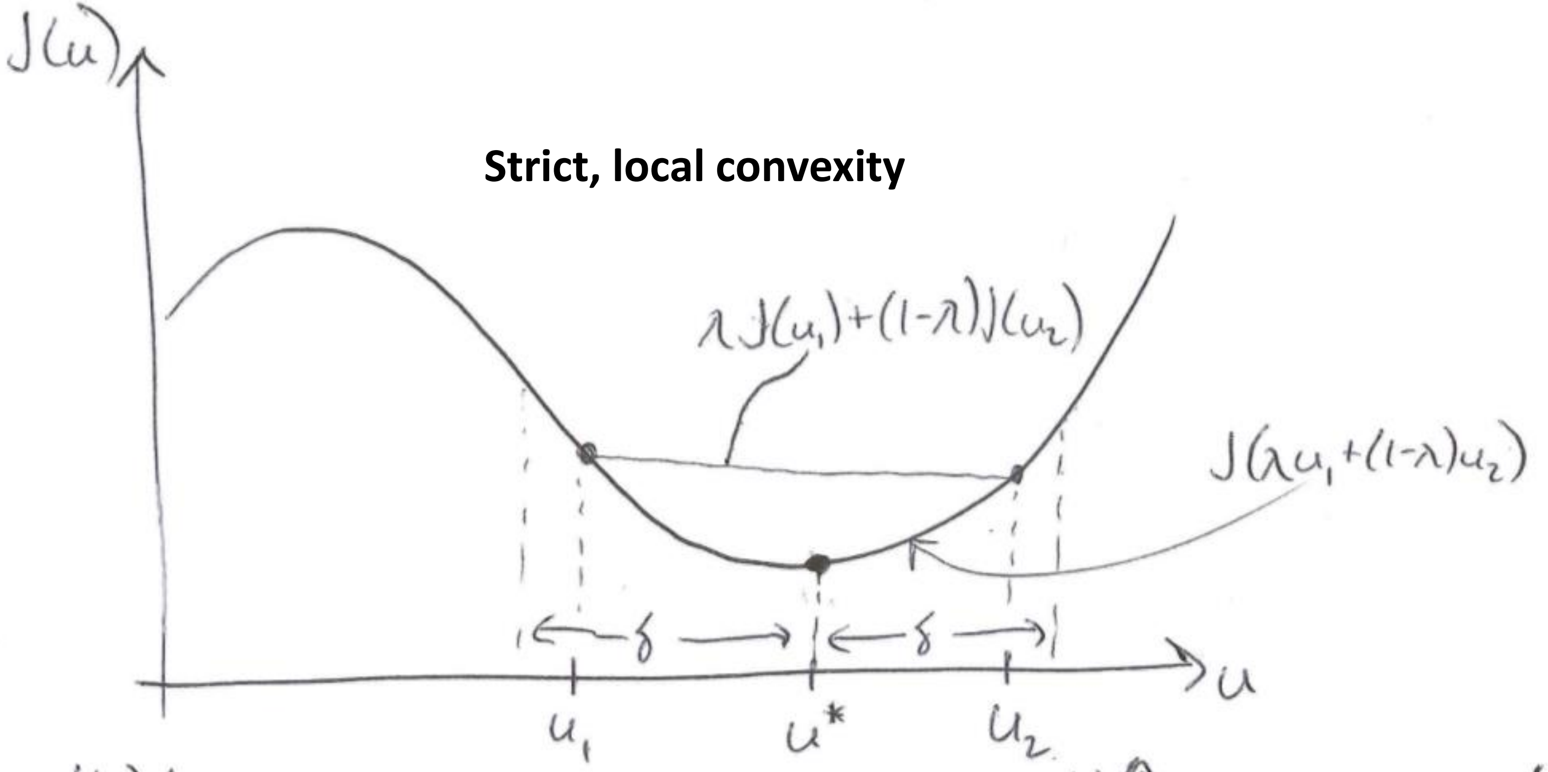
Family of valid optima ($\mathbf{u}^*$)



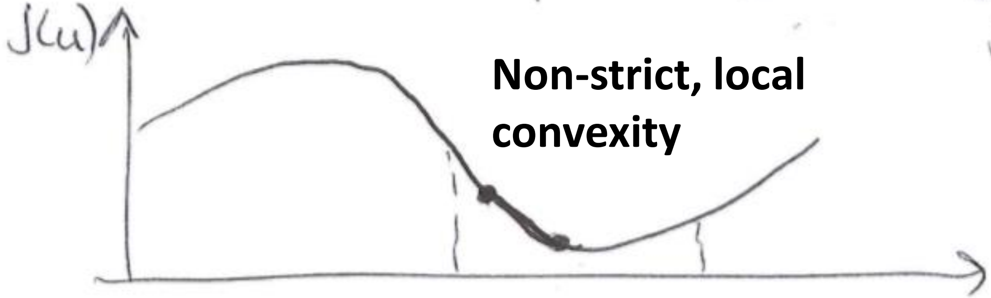
Single isolated optimum ($\mathbf{u}^*$)

Convexity – Illustrations from Class

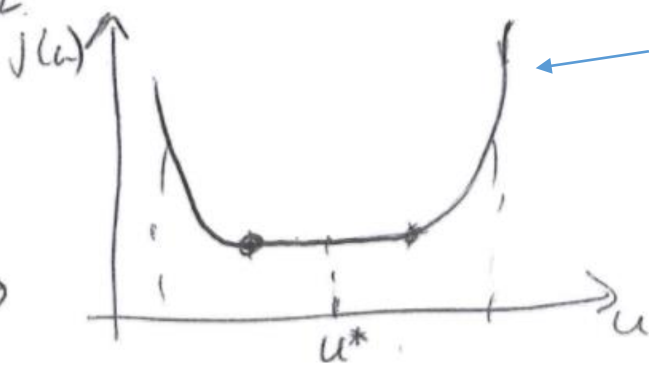
Strict, local convexity



Non-strict, local convexity



Non-strict, global Convexity (as far as we can tell)



Testing for Convexity - Scalar Case

First, consider a special case…Suppose that u is a scalar. Then $J(u)$ is locally convex around u^* if and only if $\left.\frac{d^2J}{du^2}\right|_{u^*} \geq 0$. It is locally ***strictly*** convex if $\left.\frac{d^2J}{du^2}\right|_{u^*} > 0$. If these inequalities hold everywhere, then $J(u)$ is ***globally*** convex (or strictly convex).

Simple example: Consider $J(u) = (u-7)^2$. Evaluate the convexity of $J(u)$ globally and locally around $u^* = 7$.

Testing for Convexity - Scalar Case

Ex 1: $H = 2 > 0 \Rightarrow$ strictly globally convex

$u^* = 7$ is a unique global minimizer

Testing for Convexity - Vector Case

When $\mathbf{u}$ is a vector, we can test for convexity by examining the ***Hessian*** of $J(\mathbf{u})$, which is defined as follows:

$$H(\mathbf{u}) = \begin{bmatrix} \frac{\partial^2 J}{\partial u_1^2} & \cdots & \frac{\partial^2 J}{\partial u_1 \partial u_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial u_p \partial u_1} & \cdots & \frac{\partial^2 J}{\partial u_p^2} \end{bmatrix}$$

- If $H(\mathbf{u}^*)$ is **positive semidefinite**, then $J(\mathbf{u})$ is **locally convex** around $\mathbf{u}^*$. If $H(\mathbf{u}^*)$ is **positive definite**, then $J(\mathbf{u})$ is **locally *strictly* convex** around $\mathbf{u}^*$.
- If $H(\mathbf{u})$ is **positive semidefinite for all $\mathbf{u}$**, then $J(\mathbf{u})$ is **globally convex.** If $H(\mathbf{u})$ is **positive definite for all $\mathbf{u}$**, then $J(\mathbf{u})$ is **globally convex.**

Testing for Convexity - Vector Example

Consider the function $J(\mathbf{u}) = 2u_1^2 + 3u_2^2 + 3u_1u_2 + 4u_1$.

- Denoting the ***candidate*** optimal point you identified earlier as $\mathbf{u}^*$, assess the local convexity around $J(\mathbf{u})$. What does this say about the candidate optimum?

- Is $J(\mathbf{u})$ globally convex? What does this say about the candidate optimum?

Testing for Convexity - Vector Example

Ex 2: $H = \begin{bmatrix} 4 & 3 \\ 3 & 6 \end{bmatrix}$

$\det(H) = 24 - 9 = 15 > 0$

$\underline{u}^* = [-1.6 \quad 0.8]^T$ is a unique global minimizer

Note: The hessian won't always be a matrix of constant values! In cases where it is not a matrix of constant values (i.e., when it depends on elements of $\mathbf{u}$):

- Substitute $u = u^*$ to evaluate local convexity
- The hessian must be positive (semi-)definite for all values of $\mathbf{u}$ to guarantee global convexity.

Testing for Convexity – Optimal Control Example – 5 Bonus Points on Exam 1

Consider the following discrete-time system model: $x(k+1) = x(k) + u(k)$

Objective: Minimize $J\big(\mathbf{u}; x(0)\big) = \sum_{i=0}^{2}[x(i)^2 + u(i)^2]$

Given: $x(0) = 10$

Tasks:

- Evaluate the Hessian, and assess local convexity (around the previously-determined candidate optimum) and global convexity
- What does this imply about the previously-determined candidate optimal trajectory?

Summary

Given an **unconstrained** optimization problem (minimize $J(\mathbf{u})$ subject to $\mathbf{u} \in \mathbb{R}^{\dim(\mathbf{u})}$), a **finite optimum** (minimizer) can be determined through the following procedure:

- Compute $\nabla J(\mathbf{u})$, and determine $\mathbf{u}^*$ for which $\nabla J(\mathbf{u}^*) = 0$
- Compute the Hessian to determine whether candidate $\mathbf{u}^*$ values are indeed local or global minimizers

Potential complication: Finding $\mathbf{u}^*$ for which $\nabla J(\mathbf{u}^*) = 0$ often leads to a system of nonlinear equations for which a solution is hard to obtain. In the coming lectures, we will learn about efficient numerical techniques for converging to $\mathbf{u}^*$

Preview of next lecture (and beyond)

Topics for lecture 6-7:

- Gradient-based methods for unconstrained convex optimization
- Newton’s method for unconstrained convex optimization

Beyond lectures 6-7, we will examine several different techniques for *constrained* convex optimization

- Linear and quadratic programming
- Sequential quadratic programming (SQP)
- Dynamic programming for global optimization