# ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

## Review of Unconstrained Optimization

# Constraints on the Plant Output

- The predicted plant output, $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{x}_a[k] + \boldsymbol{Z}\Delta\boldsymbol{U}$
- Suppose now that the following constraints are imposed on the predicted plant's output,

$$\boldsymbol{Y}^{\min} \leq \boldsymbol{Y} \leq \boldsymbol{Y}^{\max}$$

- Represent the above as

$$\begin{bmatrix} -\boldsymbol{Y} \\ \boldsymbol{Y} \end{bmatrix} \leq \begin{bmatrix} -\boldsymbol{Y}^{\min} \\ \boldsymbol{Y}^{\max} \end{bmatrix}$$

- Represent the above as

$$\begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{Z} \end{bmatrix} \Delta\boldsymbol{U} \leq \begin{bmatrix} -\boldsymbol{Y}^{\min} + \boldsymbol{W}\boldsymbol{x}_a[k] \\ \boldsymbol{Y}^{\max} - \boldsymbol{W}\boldsymbol{x}_a[k] \end{bmatrix}$$

- Conclusion: we need an effective method of minimizing a function of many variables, $J(\Delta\boldsymbol{U})$, subject to inequality constraints

# An Optimizer for Solving Constrained Optimization Problems

- At each sampling time the MPC calls for a solution to a constrained optimization problem of the form,

$$\begin{array}{ll} \text{minimize} & J(\Delta \boldsymbol{U}) \\ \text{subject to} & \boldsymbol{g}(\Delta \boldsymbol{U}) \leq \boldsymbol{0}, \end{array}$$

  where $\boldsymbol{g}(\Delta \boldsymbol{U}) \leq \boldsymbol{0}$ contains inequality constrains
- Need an iterative methods for solving the above optimization problems
- To proceed, we first present the descent gradient method, followed by the Newton's method, for solving unconstrained optimization problems of the form,

$$\text{minimize } J(\Delta \boldsymbol{U})$$

# Gradient of a function

- For simplicity, denote the argument of a function of many variables as $\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^N$, and denote the function being minimized as $f$, where $f : \mathbb{R}^N \to \mathbb{R}$
- The method of the gradient descent is based on the following property of the gradient of a differentiable function, $f$, on $\mathbb{R}^N$

## Theorem

*At a given point $\boldsymbol{x}^{(0)}$, the vector*

$$\boldsymbol{v} = -\nabla f\left(\boldsymbol{x}^{(0)}\right)$$

*points in the direction of most rapid decrease of $f$ and the rate of increase of $f$ at $\boldsymbol{x}^{(0)}$ in the direction $\boldsymbol{v}$ is $-\left\|\nabla f\left(\boldsymbol{x}^{(0)}\right)\right\|$, equivalently, the rate of decrease of $f$ at $\boldsymbol{x}^{(0)}$ in the direction $\boldsymbol{v}$ is $\left\|\nabla f\left(\boldsymbol{x}^{(0)}\right)\right\|$*

# Gradient Descent Method

- If we wish to minimize a differentiable function, $f$, then moving in the direction of the negative gradient is a good direction
- The gradient descent algorithm

  $$\boldsymbol{x}^{[k+1]} = \boldsymbol{x}^{[k]} - \alpha \nabla f\left(\boldsymbol{x}^{[k]}\right),$$

  where $\alpha > 0$ is a step size

# Second-Order Sufficiency Conditions for a Minimum

- Recall a well-known theorem referred to as the second-order Taylor's formula or the extended law of mean:

## Theorem

*Suppose that $f(x), f'(x), f''(x)$ exist on the closed interval $[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$. If $x^*$, $x$ are any two different points of $[a, b]$, then there exists a point $z$ strictly between $x^*$ and $x$ such that*

$$f(x) = f(x^*) + f'(x^*)(x - x^*) + \frac{f''(z)}{2}(x - x^*)^2$$

# Application of the second-order Taylor's formula

- Using the above formula we observe that if
  - $f'(x^*) = 0$, and
  - $f''(x^*) > 0$,

  then
  $$f(x) = f(x^*) + \text{ a positive number}$$
  for all $x$ "close" to $x^*$. Indeed, if $f''(x)$ is continuous at $x^*$ and $f''(x^*) > 0$, then $f''(x) > 0$ for all $x$ in some neighborhood of $x^*$. Therefore,
  $$f(x) > f(x^*) \qquad \text{for all } x \text{ close to } x^*,$$
  which means that $x^*$ is a strict local minimizer of $f$

# Extensions to functions of many variables

## Theorem

*Suppose that $\boldsymbol{x}^*$, $\boldsymbol{x}$ are points in $\mathbb{R}^N$ and that $f$ is a function of $N$ variables with continuous first and second partial derivatives on some open set containing the line segment*

$$[\boldsymbol{x}^*, \boldsymbol{x}] = \left\{\boldsymbol{w} \in \mathbb{R}^N : \boldsymbol{w} = \boldsymbol{x}^* + t\left(\boldsymbol{x} - \boldsymbol{x}^*\right); 0 \leq t \leq 1\right\}$$

*joining $\boldsymbol{x}^*$ and $\boldsymbol{x}$. Then there exists a $\boldsymbol{z} \in [\boldsymbol{x}^*, \boldsymbol{x}]$ such that*

$$f(\boldsymbol{x}) = f\left(\boldsymbol{x}^*\right) + \nabla f\left(\boldsymbol{x}^*\right)^\top \left(\boldsymbol{x} - \boldsymbol{x}^*\right) + \frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x}^*\right)^\top \boldsymbol{F}\left(\boldsymbol{z}\right)\left(\boldsymbol{x} - \boldsymbol{x}^*\right),$$

*where $\boldsymbol{F}(\cdot)$ is the Hessian of $f$, that is, the second derivative of $f$*

# Sufficient conditions for a minimizer of functions of many variables

If

- $\boldsymbol{x}^*$ is a critical point, that is, $\nabla f\left(\boldsymbol{x}^*\right) = \mathbf{0}$, and
- $\boldsymbol{F}\left(\boldsymbol{x}^*\right) > 0$,

then

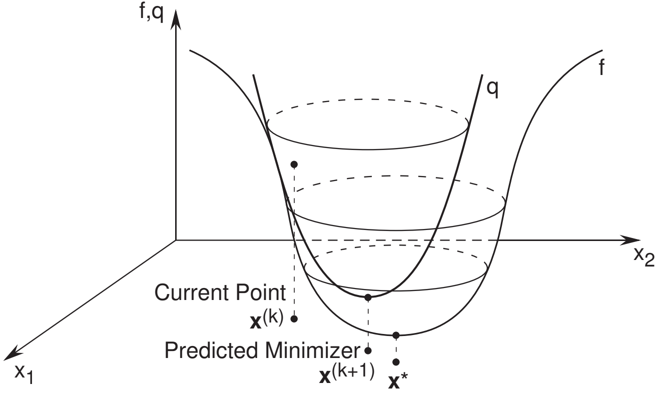
$$f(\boldsymbol{x}) = f\left(\boldsymbol{x}^*\right) + 0 + \text{a positive number}$$

for all $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{x}^*$. Therefore for all $\boldsymbol{x} \neq \boldsymbol{x}^*$ in some neighborhood of $\boldsymbol{x}^*$, we have

$$f(\boldsymbol{x}) > f\left(\boldsymbol{x}^*\right),$$

which implies that $\boldsymbol{x}^*$ is a strict local minimizer

# Newton's Method



# Idea behind Newton's method

- The idea behind Newton's method for function minimization—minimize the quadratic approximation rather than the function itself
- Newton's method seeks a critical point, $\boldsymbol{x}^*$, of a given function
- If at this critical point we have $\boldsymbol{F}(\boldsymbol{x}^*) > 0$, then $\boldsymbol{x}^*$ is a strict local minimizer of $f$
- Can obtain a quadratic approximation $q$ of $f$ at $\boldsymbol{x}^*$ from the second-order Taylor series expansion of $f$ about $\boldsymbol{x}^*$,

$$q(\boldsymbol{x}) = f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\top \boldsymbol{F}(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*)$$

- Note that

$$q(\boldsymbol{x}^*) = f(\boldsymbol{x}^*), \quad \nabla q(\boldsymbol{x}^*) = \nabla f(\boldsymbol{x}^*),$$

  as well as their Hessians, that is, their second derivatives evaluated at $\boldsymbol{x}^*$ are equal

# Newton's algorithm

- A critical point of $q$ can be obtained by solving the algebraic equation,
$$\nabla q(\boldsymbol{x}) = \boldsymbol{0},$$
that is, by solving the equation
$$\nabla q(\boldsymbol{x}) = \nabla f\left(\boldsymbol{x}^*\right) + \boldsymbol{F}\left(\boldsymbol{x}^*\right)\left(\boldsymbol{x} - \boldsymbol{x}^*\right) = \boldsymbol{0}$$
- Suppose that we have a quadratic approximation of $f$ at a point $\boldsymbol{x}^{[k]}$, that is,
$$\begin{aligned} q(\boldsymbol{x}) &= f\left(\boldsymbol{x}^{[k]}\right) + \nabla f\left(\boldsymbol{x}^{[k]}\right)^\top \left(\boldsymbol{x} - \boldsymbol{x}^{[k]}\right) \\ &\quad + \frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x}^{[k]}\right)^\top \boldsymbol{F}\left(\boldsymbol{x}^{[k]}\right)\left(\boldsymbol{x} - \boldsymbol{x}^{[k]}\right) \end{aligned}$$
- Assume that $\det \boldsymbol{F}\left(\boldsymbol{x}^{[k]}\right) \neq 0$

# The Newton's method for minimizing a function of many variables

- Denote by $\boldsymbol{x}^{[k+1]}$ the solution to

$$\nabla q(\boldsymbol{x}) = \nabla f\left(\boldsymbol{x}^{[k]}\right) + \boldsymbol{F}\left(\boldsymbol{x}^{[k]}\right)\left(\boldsymbol{x} - \boldsymbol{x}^{[k]}\right) = \mathbf{0}$$

- The Newton's method for minimizing a function of many variables $f$

$$\boldsymbol{x}^{[k+1]} = \boldsymbol{x}^{[k]} - \boldsymbol{F}\left(\boldsymbol{x}^{[k]}\right)^{-1} \nabla f\left(\boldsymbol{x}^{[k]}\right)$$

- Note that $\boldsymbol{x}^{[k+1]}$ is a critical point of the quadratic function $q$ that approximates $f$ at $\boldsymbol{x}^{[k]}$
- Computationally efficient representation of Newton's algorithm

$$\boldsymbol{x}^{[k+1]} = \boldsymbol{x}^{[k]} - \Delta \boldsymbol{x}^{[k]},$$

where $\Delta \boldsymbol{x}^{[k]}$ is obtained by solving

$$\boldsymbol{F}\left(\boldsymbol{x}^{[k]}\right) \Delta \boldsymbol{x}^{[k]} = \nabla f\left(\boldsymbol{x}^{[k]}\right)$$