Lecture 13

Optimization, Null-Finding, and Control

References

http://en.wikipedia.org/wiki/Optimization
http://en.wikipedia.org/wiki/Gradient_descent
http://en.wikipedia.org/wiki/Newton%27s_method
http://en.wikipedia.org/wiki/Broyden%27s_method
http://en.wikipedia.org/wiki/Constrained_optimization
http://en.wikipedia.org/wiki/Quadratic_programming
http://en.wikipedia.org/wiki/Linear_programming
http://www.mathworks.com/products/optimization/
http://en.wikipedia.org/wiki/Control_system
http://en.wikipedia.org/wiki/Control_system_(disambiguation)
http://www.mathworks.com/help/toolbox/control/
ROOT FINDING:

Ranks of $\mathbf{F}(\mathbf{x})$, where $\mathbf{x}^2 = (x_1, x_2, \ldots, x_n)$, $\mathbf{F} = (F_1, F_2, \ldots, F_m)$:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad n$$

$$\begin{cases} F_1 (x_1, x_2, \ldots, x_n) = 0 \\ F_2 (x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ F_m (x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

$m$ equations in $n$ variables

Solution is unique, with a single root $\mathbf{x}^2$, when $\mathbf{F}$ is linear and full-rank.

In general, multiple solutions exist.
UNCONSTRAINED OPTIMIZATION:

Extrema of \( m(\hat{x}) = m(x_1, x_2, \ldots, x_n) \):

- \( \max_{x_1, \ldots, x_n} m(x_1, x_2, \ldots, x_n) \)
- \( \min_{x_1, \ldots, x_n} m(x_1, x_2, \ldots, x_n) \) \( = \max_{x_1, \ldots, x_n} [-m(x_1, \ldots, x_n)] \)

Let: \( \vec{F} = \vec{\nabla} m \), gradient of \( m \) : \( F_i = \frac{\partial m}{\partial x_i} \)

(\( \vec{H} \) Hessian of \( m \) : \( H_{ij} = \frac{\partial^2 m}{\partial x_i \partial x_j} = \frac{\partial F_i}{\partial x_j} = \frac{\partial F_j}{\partial x_i} \)

- STATIONARY POINTS: \( \hat{x} \) for which \( \vec{F}(\hat{x}) = \vec{0} \)

Necessary but not sufficient condition for (local) extrema!

\[ m \uparrow \quad \text{maxima} \]

\[ m \downarrow \quad \text{minima} \]

- LOCAL MAXIMA: Stationary points for which
  \( \vec{H}(\hat{x}) \) negative definite, or \( \Delta \hat{x}^T \vec{H} \Delta \hat{x} \leq 0, \forall \Delta \hat{x} \)

- LOCAL MINIMA: Stationary points for which
  \( \vec{H}(\hat{x}) \) positive definite, or \( \Delta \hat{x}^T \vec{H} \Delta \hat{x} \geq 0, \forall \Delta \hat{x} \)
Taylor series expansion around $\bar{x}:

$$
\mu(x^2 + \Delta x^2) = \mu(\bar{x}) + \sum_{i} \frac{\partial \mu}{\partial x_i} \Delta x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 \mu}{\partial x_i \partial x_j} \Delta x_i \Delta x_j + \ldots
$$

\[\text{higher order terms:}\]
\[\text{neglected for } \Delta x \text{ "small"}\]

\[\begin{align*}
&\mu \text{ for stationary point} \\
&\geq 0 \text{ for } \bar{\mu} \text{ pos. def.} \\
&\leq 0 \text{ for } \bar{\mu} \text{ neg. def.}
\end{align*}\]

\[\implies \text{ local maxima } \mu(x^2 + \Delta x^2) \leq \mu(\bar{x}) \text{ for } \bar{\mu} \text{ neg. def.} \\
\text{minima } \geq \]

Notes:
- $\bar{\mu}$ is symmetric, and so all eigenvalues are REAL
- $\bar{\mu}$ is POS. DEF. if all eigenvalues are POSITIVE
- $\bar{\mu}$ is NEG. if all eigenvalues are NEGATIVE
GRADIENT ASCENT / DESCENT:

iteratively searches for MAX/MIN extrema of $u(x)$:

- Initial guess $x_0$
- Iterate, $i \rightarrow i+1$:
  
  $$x_{i+1} \leftarrow x_i + \mu \frac{\partial}{\partial x_i} f(x_i)$$

If $\mu$ is "sufficiently" small, the series $x_i$ converge to a local maximum ($\mu > 0$) or minimum ($\mu < 0$) of $u(x)$:

$$u(x_{i+1}) = u(x_i + \mu \frac{\partial}{\partial x_i} f(x_i)) \approx u(x_i) + \frac{\partial}{\partial x_i} f(x_i) + \frac{1}{2} \mu^2 \nabla^2 u(x_i)$$

$$\Rightarrow u(x_{i+1}) \approx u(x_i) + \mu \|F\|^2 \geq u(x_i) \quad \text{for} \quad \mu > 0 \text{ and small}$$

Equality: at convergence: $F = 0$, and $\nabla^2 u$ POS. DEF. NEgm.

If $\mu$ is too large then series diverges...
NEWTON'S METHOD:

A second-order null-finding/optimization method for faster convergence than first-order methods such as gradient ascent/descent:

\[
\overrightarrow{\mathbf{F}}(\overrightarrow{x}_{i+1}) = \overrightarrow{\mathbf{F}}(\overrightarrow{x}_i + \Delta \overrightarrow{x}_i) \approx \overrightarrow{\mathbf{F}}(\overrightarrow{x}_i) + \overrightarrow{\nabla} \cdot \overrightarrow{H} \cdot \Delta \overrightarrow{x}_i + \ldots
\]

\[
(\overrightarrow{x}_{i+1} - \overrightarrow{x}_i) = \overrightarrow{0} \text{ for STATIONARY POINT}
\]

\[
\Rightarrow \quad \overrightarrow{x}_{i+1} \leftarrow \overrightarrow{x}_i - (\overrightarrow{H}(\overrightarrow{x}_i))^{-1} \overrightarrow{\nabla} \overrightarrow{F}(\overrightarrow{x}_i)
\]

**in inverse Hessian, rather than constant scalar \( \mu \)**

*e.g., in 1-D:*

\[
X_{i+1} \leftarrow X_i - \frac{1}{h(X_i)} \cdot f(X_i) \quad \text{where} \quad h(X) = \frac{df(X)}{dx}
\]
SECANT METHOD:

A second-order null-finding/optimization method with a linear approximation for \( f \), and values for \( f \) only:

- In 1-D:

\[
h_i(x_i) = \frac{df(x_i)}{dx_i} \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}
\]

\[
\Rightarrow x_{i+1} \leftarrow x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} \cdot f(x_i)
\]

![Graph of function](image)

- Requires two starting points \( x_0, x_1 \)

- In higher dimensions, n-D: more involved

Broyden - Fletcher - Goldfarb - Shanno (BFGS) method
CONSTRANDED OPTIMIZATION:

Find extrema (maxima) of $\mu(x^2)$ subject to equality and/or inequality constraints:

$$\max \quad \mu(x^2) \quad \text{subject to} \quad \begin{cases} \forall i, j \in \{1, \ldots, p, q\} \\ x_1, x_2, \ldots, x_m \\ \sum_{i=1}^{p} \lambda_i \sigma_i(x^2) + \sum_{j=1}^{q} \mu_j \cdot \psi_j(x^2) \end{cases}$$

When $\mu(x^2)$ is concave in $x^2$, and $\sigma_i(x^2)$ and $\psi_j(x^2)$ are linear in $x^2$, then sufficient conditions for an optimum $x^2$ are given by the Karush-Kuhn-Tucker (KKT) conditions:

- Stationarity: $\nabla (\mu(x^2) + \sum_{i=1}^{p} \lambda_i \sigma_i(x^2) + \sum_{j=1}^{q} \mu_j \cdot \psi_j(x^2)) = 0$

- Feasibility & slacksness:

  \begin{align*}
  \sigma_i(x^2) &= C_i, \quad \forall i = 1, \ldots, p \\
  \psi_j(x^2) &\leq d_j \quad \text{and} \quad \mu_j \cdot (\psi_j(x^2) - d_j) = 0, \quad \forall j = 1, \ldots, q \\
  \mu_j &\geq 0
  \end{align*}

Examples:

- $\mu(x^2) = A \cdot x \quad \rightarrow \quad \text{LINEAR PROGRAMMING}$

- $\mu(x^2) = \frac{1}{2} x^T \Sigma x \quad \rightarrow \quad \text{QUADRATIC PROGRAMMING}$

See Matlab toolboxes
LINEAR CONTROL : A cursory introduction

\[ U \xrightarrow{} H(j\omega) \xrightarrow{} X \]

CONTROL INPUT
( insulin release, ... )

"PLANT"
( body, instrument, ... )

OUTPUT
( glucose level, ... )

A controller will attempt to control the input \( U \) to drive the output \( X \) towards a desired state \( Y \):

\[
\begin{align*}
X(j\omega) &= H(j\omega) \cdot U(j\omega) \\
U(j\omega) &= F(j\omega) \cdot (Y(j\omega) - X(j\omega))
\end{align*}
\]

\[
\implies X(j\omega) = \frac{F(j\omega) H(j\omega)}{1 + F(j\omega) H(j\omega)} Y(j\omega)
\]

or \[
X(j\omega) = \frac{1}{1 + \frac{1}{F(j\omega) H(j\omega)}} Y(j\omega) \quad \text{for} \quad |FH| \gg 1
\]
Considerations:

1. **STABILITY**: The control loop becomes unstable when the open loop gain $F(j\omega)H(j\omega)$ reaches $-1$.

   - **ABSOLUTE STABILITY** as long as the open loop phase $\angle F(j\omega)H(j\omega)$ is greater than $-180^\circ$ (and less than $180^\circ$) for all frequencies where the open loop gain $|F(j\omega)H(j\omega)|$ is greater than 1.

   - **PHASE MARGIN** needs to be positive: (and ideally $\geq 60^\circ$) to minimize ringing.
2. CONTROLLER DESIGN: A universal design that often works is the P.I.D. (proportional - integral - differential) controller:

\[ F(jw) = a + b \frac{1}{jw} + c \cdot jw, \quad a \text{ "P"}, \quad b \text{ "I"}, \quad c \text{ "D"} \]

\[ u(t) = a \cdot (y(t) - x(t)) + b \int_{-\infty}^{t} (y(\tau) - x(\tau)) d\tau + c \left( \frac{dy}{dt} - \frac{dx}{dt} \right) \]

\[ \frac{|F(jw)|}{\omega} \quad \text{\( (\log) \)} \]

\[ \angle F(jw) \quad \text{\( 90^\circ \)} \]

\[ w = 2\pi f \]

\[ \frac{b}{c} \quad \frac{d}{c} \]

\[ (a^2 > bc) \]
3. DISCRETE TIME SYSTEMS: Some principles, using
Z-transforms rather than Fourier/Laplace
\[ Z = e^{ST} = e^{j\omega T} \]
where \( T \) is time step
(unit time advance)
\( \frac{1}{T} \) is sampling rate

\[ u(t) \to u[n] = u(mT) \]
\[ x(t) \to x[n] = x(mT) \]
\[ y(t) \to y[n] = y(mT) \]

4. NONLINEAR SYSTEMS: Complex!
Sometimes a Taylor expansion of \( H \) (and \( F \))
around a known stable state (or limit cycle) of
\( U, X \) and \( Y \) will work.

\[ U = U_0 + \tilde{U}(j\omega) \quad \text{with} \quad |\tilde{U}| \ll |U_0| \]
\[ X = X_0 + \tilde{X}(j\omega) \quad \text{with} \quad |\tilde{X}| \ll |X_0| \]
\[ X = H(U) = H(U_0 + \tilde{U}) \approx X_0 + \frac{\partial H}{\partial U} \mid_{U_0} \cdot \tilde{U} \]
(nonlinear dynamic system)

\[ \Rightarrow \tilde{X}(j\omega) \approx \tilde{H}(j\omega) \cdot \tilde{U}(j\omega) \quad \text{with} \quad \tilde{H}(j\omega) = \frac{\partial H}{\partial U} \mid_{U_0} \]
where \( \frac{\partial}{\partial t} \to j\omega \)