Path Constraints and Numerical Optimization

Robert Stengel
Optimal Control and Estimation, MAE 546, Princeton University, 2015

- State and Control Equality Constraints
- Pseudoinverse
- State and Control Inequality Constraints
- Numerical Methods
  - Penalty function and collocation
  - Neighboring optimal
  - Quasilinearization
  - Gradient
    - Newton-Raphson
    - Steepest descent
- Optimal Treatment of an Infection

Copyright 2015 by Robert Stengel. All rights reserved. For educational use only.
http://www.princeton.edu/~stengel/MAE546.html
http://www.princeton.edu/~stengel/OptConEst.html

Minimization with Equality Constraints
Minimization with Equality Constraint on State and Control

\[
\min_{u(t)} J = \phi [x(t_f)] + \int_{t_a}^{t_f} L [x(t), u(t)] \, dt
\]

- subject to
  - **Dynamic Constraint**
    \[
    \dot{x}(t) = f[x(t), u(t)], \quad x(t_o) \text{ given}
    \]
  - **State/Control Equality Constraint**
    \[
    c[x(t), u(t)] = 0 \quad \text{in } (t_o, t_f); \quad \dim(c) = (r \times 1) \leq (m \times 1)
    \]

**Aeronautical Example:**
Longitudinal Point-Mass Dynamics

\[
\begin{align*}
\dot{V} &= \left( T - C_D \frac{1}{2} \rho V^2 S \right) / m - g \sin \gamma \\
\dot{\gamma} &= \frac{1}{V} \left[ \left( C_L \frac{1}{2} \rho V^2 S \right) / m - g \cos \gamma \right] \\
\dot{h} &= V \sin \gamma \\
\dot{r} &= V \cos \gamma \\
\dot{m} &= -(SFC)(T)
\end{align*}
\]

- \( x_1 = V \): Velocity, m/s
- \( x_2 = \gamma \): Flight path angle, rad
- \( x_3 = h \): Height, m
- \( x_4 = r \): Range, m
- \( x_5 = m \): Mass, kg
Aeronautical Example:
Longitudinal Point-Mass Dynamics

\[ u_1 = \delta T : \text{Throttle setting, } \% \]
\[ u_2 = \alpha : \text{Angle of attack, rad} \]

\[ T = T_{\text{max}} (e^{-\beta u}) \delta T : \text{Thrust, N} \]
\[ C_D = \left( C_{D_0} + \epsilon C_{L}^2 \right) \text{Drag coefficient} \]
\[ C_L = C_{L_{\alpha}} \alpha \text{Lift coefficient} \]
\[ S = \text{Reference area, m}^2 \]
\[ m = \text{Vehicle mass, kg} \]
\[ \rho = \text{Air density} = \rho_{SL} e^{-\beta u}, \text{kg/m}^3 \]
\[ g = \text{Gravitational acceleration, m/s}^2 \]
\[ SFC = \text{Specific Fuel Consumption, g/kN-s} \]

Path Constraint Included in the Cost Function Hamiltonian

Constraint must be satisfied at every instant of the trajectory
Dimension of the constraint \( \leq \) dimension of the control

\[ J_1 = \psi \left[ x(t_f) \right] + \int_{t_0}^{t_f} \left\{ L + \lambda_i^T (t) [ f - x(t) ] + \mu^T c \right\} dt \]

\[ c \left[ x(t), u(t) \right] \equiv 0 \text{ in } \left( t_0, t_f \right) \]

The constraint is adjoined to the Hamiltonian

\[ H \triangleq L + \lambda_i^T f + \mu^T c \]

\[ \text{dim}(x) = \text{dim}(f) = \text{dim}(\lambda) = n \times 1 \]
\[ \text{dim}(u) = m \times 1 \]
\[ \text{dim}(c) = \text{dim}(\mu) = r \times 1, r \leq m \]
Euler-Lagrange Equations Including Equality Constraint

\[ \lambda(t_f) = \left\{ \frac{\partial H(x(t_f),x(t),u(t),\lambda,t)}{\partial x} \right\}^T \]

\[ \lambda = -\left\{ \frac{\partial H(x,u,\lambda,c,\mu,t)}{\partial x} \right\}^T = -\left[ \frac{\partial L}{\partial x} + \lambda^T \frac{\partial f}{\partial x} + \mu^T \frac{\partial c}{\partial x} \right] = -\left[ L^T + F^T \lambda + \mu^T G^T \lambda \right] \mu = 0 \]

\[ c \left[ x(t),u(t) \right] \equiv 0 \text{ in } (t_o,t_f) \]

**No Optimization When** \( r = m \)

- Control entirely specified by constraint
  - \( m \) unknowns, \( m \) equations

\[ c \left[ x(t),u(t) \right] \equiv 0 \Rightarrow u(t) = fcn \left[ x(t) \right] \]

**Example**

\( c = Ax(t) + Bu(t) = 0; \quad \text{dim}(x) = n \times 1; \quad \text{dim}(c) = \text{dim}(u) = m \times 1 \)

\( \text{dim}(A) = m \times n; \quad \text{dim}(B) = m \times m \)

\( u(t) = -B^{-1}Ax(t) \)

- Constraint Lagrange multiplier is irrelevant but can be expressed
  - from \( dH/d\mu = 0, \)

\[ \mu = -\left( \frac{\partial c}{\partial u} \right)^T \left[ \frac{\partial L}{\partial u} \right] + G^T \lambda \]

\( \frac{\partial c}{\partial u} \) is square and non-singular
No Optimization When $r = m$

**MAINTAIN CONSTANT VELOCITY AND FLIGHT PATH ANGLE**

$$c[x(t), u(t)] \equiv 0 = \begin{bmatrix} 0 = \dot{V} = \frac{T_{\text{max}} \delta T - (C_{D_0} + \varepsilon C_L^2) \frac{1}{2} \rho V^2 S}{m - g \sin \gamma} \\ 0 = \dot{\gamma} = \frac{1}{V} \left[ \left( C_{r_0} \alpha \frac{1}{2} \rho V^2 S \right) / m - g \cos \gamma \right] \end{bmatrix}$$

$$\Rightarrow u(t) = f(t) \equiv [x(t)]$$

**Effect of Constraint Dimensionality:**

$r < m$

**MINIMIZE FUEL AND CONTROL USE WHILE MAINTAINING CONSTANT FLIGHT PATH ANGLE**

$$\min_{u(t)} J = \int_{t_0}^{t_f} \left( q \dot{m}^2 + r_1 \dot{u}_1^2 + r_2 \dot{u}_2^2 \right) dt$$

$$c[x(t), u(t)] \equiv 0 = \dot{\gamma} = \frac{1}{V(t)} \left[ \left( C_{L_0} \alpha \frac{1}{2} \rho V^2(t) S \right) / m - g \cos \gamma(t) \right]$$

$$\gamma(0) = \gamma_{\text{desired}}$$
Effect of Constraint Dimensionality: \( r < m \)

\[
\begin{align*}
\left( \frac{\partial c}{\partial u} \right) & \text{ is not square when } r < m \\
\therefore \left( \frac{\partial c}{\partial u} \right) & \text{ is not strictly invertible}
\end{align*}
\]

- Three approaches to constrained optimization
  - Algebraic solution for \( r \) control variables using an invertible subset of the constraint
  - Pseudoinverse of control effect
  - "Soft" constraint

Effect of Constraint Dimensionality: \( r < m \)

Algebraic solution for \( r \) control variables using an invertible subset of the constraint

**Example 1**

\[
\begin{align*}
\dim(x) &= n \times 1; \quad \dim(A_r) = r \times n \\
\dim(u) &= m \times 1; \quad \dim(u_r) = r \times 1; \quad \dim(B_r) = r \times r \\
c &= A\times(t) + B_ru_r(t) = 0; \quad \det(B_r) \neq 0 \\
u_r(t) &= -B_r^{-1}A\times(t)
\end{align*}
\]

**Example 2**

\[
\begin{align*}
\dim(u) &= m \times 1; \quad \dim(u_r) = r \times 1; \quad \dim(B_2) = r \times r \\
c &= \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} u_r \\ u_{uv-r} \end{bmatrix} = B_1u_r + B_2u_{uv-r} = 0; \quad \det(B_2) \neq 0 \\
u_r(t) &= -B_2^{-1}B_1u_{uv-r}(t)
\end{align*}
\]
Second Approach: Satisfy Constraint
Using Left Pseudoinverse: \( r < m \)

\[
\left[ \left( \frac{\partial c}{\partial u} \right)^T \right]_L \text{ is the left pseudoinverse of control sensitivity}
\]

\[
\dim \left[ \left( \frac{\partial c}{\partial u} \right)^T \right]_L = r \times m
\]

Lagrange multiplier

\[
\mu_L = -\left[ \left( \frac{\partial c}{\partial u} \right)^T \right]_L \left[ \left( \frac{\partial L}{\partial u} \right)^T + G^T \lambda \right]
\]

Pseudoinverse of Matrix

\[
y = Ax
\]

\[
(m \times 1) = (m \times r)(r \times 1)
\]

\[
dim(x) = r \times 1 \quad dim(y) = m \times 1
\]

\[r = m, \text{ A is square and non-singular}\]

\[
x = A^{-1}y
\]

\[
(r \times 1) = (r \times m)(m \times 1) = (r \times r)(r \times 1)
\]

\[r \neq m, \text{ A is not square}\]

Use pseudoinverse of A

\[
x = A^\dagger y
\]

\[
(r \times 1) = (r \times m)(m \times 1)
\]

Maximum rank of A is \( r \) or \( m \), whichever is smaller

See http://en.wikipedia.org/wiki/Moore-Penrose_pseudoinverse
Left Pseudoinverse

Maximum rank of A is $r$ or $m$, whichever is smaller

$r < m$, Left pseudoinverse is appropriate

\[
\begin{align*}
Ax &= y \\
A^T Ax &= A^T y \\
A_L &= \left(A^T A \right)^{-1} A^T \\
x &= A_L y
\end{align*}
\]

Averaging solution

\[
\begin{align*}
dim(x) &= r \times 1 \\
dim(y) &= m \times 1 \\
dim(A^T A) &= r \times r \\
dim(AA^T) &= m \times m
\end{align*}
\]

Right Pseudoinverse

$r > m$, Right pseudoinverse is appropriate

\[
\begin{align*}
Ax &= y = I y \\
Ax &= \left(AA^T \right)\left(AA^T \right)^{-1} y \\
&= I y \\
A_R &= \left(A^T A \right)^{-1} A^T \\
x &= A_R y
\end{align*}
\]

Minimum Euclidean error norm solution

\[
\begin{align*}
dim(x) &= r \times 1 \\
dim(y) &= m \times 1 \\
dim(A^T A) &= r \times r \\
dim(AA^T) &= m \times m
\end{align*}
\]
Left Pseudoinverse Example

\[ \mathbf{Ax} = \mathbf{y}, \quad r < m \]
\[ \mathbf{x} = (\mathbf{A}^\mathsf{T} \mathbf{A})^{-1} \mathbf{A}^\mathsf{T} \mathbf{y} \]
\[
\begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix}
\mathbf{x} =
\begin{bmatrix}
  2 \\
  6 \\
\end{bmatrix}
\]
\[
\mathbf{x} = \left( \begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix} \begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix} \right)^{-1} \begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix}
\begin{bmatrix}
  2 \\
  6 \\
\end{bmatrix}
\]
\[
= \frac{1}{10} (20) = 2
\]

Unique solution

Right Pseudoinverse Example

\[ \mathbf{Ax} = \mathbf{y}, \quad r > m \]
\[ \mathbf{x} = \mathbf{A}^\mathsf{T} \left( \mathbf{A} \mathbf{A}^\mathsf{T} \right)^{-1} \mathbf{y} \]
\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
\end{bmatrix}
= \left( \begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix} \begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix} \begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix} \right)^{-1} \begin{bmatrix}
  1 \\
  3 \\
\end{bmatrix}
\begin{bmatrix}
  14 \\
\end{bmatrix}
\]
\[
= \frac{1}{3} (14) = \begin{bmatrix}
  1.4 \\
  4.2 \\
\end{bmatrix}
\]

At least two solutions
\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
\end{bmatrix}
= \begin{bmatrix}
  2 \\
  4 \\
\end{bmatrix}
\]
satisfies the equation, but \( \| \mathbf{x} \| = \sqrt{20} \)
\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
\end{bmatrix}
= \begin{bmatrix}
  1.4 \\
  4.2 \\
\end{bmatrix}
\]
and \( \| \mathbf{x} \| = \sqrt{19.6} \)

Minimum - norm solution
Necessary Conditions Use Left Pseudoinverse for $r < m$

Optimality conditions

$$
\lambda(t_f) = \left[ \frac{\partial \phi(x(t_f))}{\partial x} \right]^T
$$

$$
\dot{\lambda} = - \left[ L^T_x + F^T \lambda + \left( \frac{\partial c}{\partial x} \right)^T \mu \right]
$$

$$
\left[ \left( \frac{\partial L}{\partial u} \right)^T + G^T \lambda + \left( \frac{\partial c}{\partial u} \right)^T \mu \right] = 0
$$

with

$$
c [x(t), u(t)] \equiv 0
$$

Third Approach: Penalty Function Provides “Soft” State-Control Equality Constraint: $r < m$

$$
L \triangleq L_{original} + \varepsilon c^T c
$$

$\varepsilon$ : Scalar penalty weight

Euler-Lagrange equations are adjusted accordingly

$$
\lambda(t_f) = \left[ \frac{\partial \phi(x(t_f))}{\partial x} \right]^T
$$

$$
\dot{\lambda} = - \left[ L^T_x + F^T \lambda \right]
$$

$$
\left[ \left( \frac{\partial L_{orig}}{\partial u} + 2 \varepsilon c^T \frac{\partial c}{\partial u} \right)^T + G^T \lambda \right] = 0
$$

$$
c [x(t), u(t)] \equiv 0
$$
Equality Constraint on State Alone

\[ c[x(t)] = 0 \quad \text{in } (t_o, t_f) \]

\[ J_1 = \psi[x(t_f)] + \int_{t_o}^{t_f} \left\{ L + \lambda_1^T(t)[f - x(t)] + \mu^T c \right\} dt \]

Hamiltonian

\[ H \triangleq L + \lambda_1^T f + \mu^T c \]

Constraint is insensitive to control perturbations to first order

\[ \Delta c = \left( \frac{\partial c}{\partial x} \right) \Delta x + \left( \frac{\partial c}{\partial u} \right) \Delta u = 0 \]

Example of Equality Constraint on State Alone

MINIMIZE FUEL AND CONTROL USE WHILE MAINTAINING CONSTANT ALTITUDE

\[ \min_{u(t)} J = \int_{t_0}^{t_f} \left( q \dot{m}^2 + r_1 u_1^2 + r_2 u_2^2 \right) dt \]

\[ c[x(t), u(t)] = c[x(t)] = 0 = h(t) - h_{desired} \]
Introduce Time-Derivative of Equality Constraint

Equality constraint has no effect on optimality condition

\[
\begin{align*}
\left\{ \frac{\partial H}{\partial u} \right\}^T &= -\left[ \left( \frac{\partial L}{\partial u} \right)^T + G^T \lambda + \left( \frac{\partial c}{\partial u} \right)^T \mu \right] \\
&= -\left[ \left( \frac{\partial L}{\partial u} \right)^T + G^T \lambda \right]
\end{align*}
\]

Solution: Incorporate time-derivative of \( c[x(t)] \) in optimization

Introduce Time-Derivative of Equality Constraint

Define \( c[x(t)] \) as the zeroth-order equality constraint

\[
c\left[ x(t) \right] \overset{\Delta}{=} c^{(0)}\left[ x(t) \right] \equiv 0
\]

Compute first-order equality constraint

\[
\frac{dc^{(0)}[x(t)]}{dt} = \frac{\partial c^{(0)}[x(t)]}{\partial t} + \frac{\partial c^{(0)}[x(t)]}{\partial x} f[x(t), u(t)]
\]

\[
\overset{\Delta}{=} c^{(1)}[x(t), u(t)] = 0
\]
Time-Derivative of Equality Constraint

Optimality condition now includes derivative of equality constraint

\[
\left[ \frac{\partial H}{\partial u} \right]^T = -\left[ \left( \frac{\partial L}{\partial u} \right)^T + G^T \lambda + \left( \frac{\partial c^{(1)}}{\partial u} \right)^T \right] \mu
\]

Subject to

\[
c^{(0)} \left[ x(t_o) \right] \equiv 0 \text{ or } c^{(0)} \left[ x(t_f) \right] \equiv 0
\]

- With equality constraint satisfied at beginning or end of trajectory, \( c^{(1)} = 0 \) assures that constraint is satisfied throughout
- If \( \partial c^{(0)}/\partial u = 0 \), differentiate again, and again, ...

State Equality Constraint Example

\[
c \left[ x(t) \right] \triangleq c^{(0)} \left[ x(t) \right] = 0 = h(t) - h_{\text{desired}}
\]

No control in the constraint; differentiate

\[
\frac{dc^{(0)} \left[ x(t) \right]}{dt} = \frac{\partial c^{(0)} \left[ x(t) \right]}{\partial t} + \frac{\partial c^{(0)} \left[ x(t) \right]}{\partial x} f \left[ x(t), u(t) \right]
\]

\[
\triangleq c^{(1)} \left[ x(t) \right] = 0 = \dot{h}(t) = V(t) \sin \gamma(t)
\]

Still no control in the constraint; differentiate again...
State Equality Constraint Example

Still no control in the constraint; differentiate again

\[
\frac{d c^{(1)}[x(t)]}{dt} = \frac{\partial c^{(1)}[x(t)]}{\partial t} + \frac{\partial c^{(1)}[x(t)]}{\partial x} f[x(t), u(t)]
\]

\[
\Delta \quad c^{(2)}[x(t)] = 0 = \ddot{h}(t) = \frac{dV}{dt}(t)\sin \gamma(t) + V(t) \frac{d}{dt}\left[\sin \gamma(t)\right]
\]

\[
= \left[\left(T_{max}\ v_0^2 - C_D \ 0.5 \rho \ V^2 \ S \right)\right] / m - g \sin \gamma(t)
\]

\[
+ \cos \gamma(t) \left[\left(C_L \ 0.5 \ rho \ V^2 \ S \right) / m - g \cos \gamma(t)\right]
\]

State Equality Constraint Example

- Control appears in the 2\textsuperscript{nd}-order equality constraint

\[
c^{(2)}[x(t), u(t)] = 0
\]

\[
= \left[\left(T_{max}\ v_0^2 - C_D \ 0.5 \rho \ V^2 \ S \right)\right] / m(t) - g \sin \gamma(t)
\]

\[
+ \cos \gamma(t) \left[\left(C_L \ 0.5 \ rho \ V^2 \ S \right) / m(t) - g \cos \gamma(t)\right]
\]

\[
H \triangleq L + \lambda_1^T f + \mu^T c^{(2)}
\]

- 0\textsuperscript{th}- and 1\textsuperscript{st}-order constraints satisfied at some point on the trajectory (e.g., \( t_0 \))

\[
c^{(0)}[x(t_0)] = 0 \quad \Rightarrow \quad h(t_0) = h_{\text{desired}}
\]

\[
c^{(1)}[x(t_0)] = 0 \quad \Rightarrow \quad \gamma(t_0) = 0
\]
Minimization with Inequality Constraints

“Hard” Inequality Constraints

Inequality constraints on state and control in \((t_0, t_f)\):

- Principal difference from equality constraints:
  - Inequality constraint normally not invoked throughout interval \((t_0, t_f)\):
    - \(c[x^*] = 0\)
    - Constrained area
    - Unconstrained area
Inequality Constraints

Weierstrass-Erdmann Conditions

- $c^T \left[ x(t), u(t), t \right] \leq 0$
- $c^T \left[ y(t), t \right] \leq 0$
- Junction between unconstrained and constrained
  "Arc" = LERTRY POINT at $t = T_1$

Transversality Conditions

- $\lambda(t^-) = \lambda(t^+)$
- $\mu(t^-) = \mu(t^+)$
- $\mu_u(t^-) = \mu_u(t^+) \neq 0$
Inequality Constraints

**“Soft” Control Inequality Constraint**

\[ L \triangleq L_{\text{original}} + \varepsilon c^T c \]

\( \varepsilon \): Scalar penalty weight

**Scalar Example**

\[ c[u(t)] = \begin{cases} 
(u - u_{\text{max}})^2, & u \geq u_{\text{max}} \\
0, & u_{\text{min}} < u < u_{\text{max}} \\
(u - u_{\text{min}})^2, & u \leq u_{\text{min}}
\end{cases} \]
# Numerical Optimization

## Numerical Optimization Methods

<table>
<thead>
<tr>
<th>Optimal Solution</th>
<th>Solution Method</th>
<th>Iteration Variables</th>
<th>Order of ODE Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>approximate ODE$^a$ $I^b$ $u(k_n,t)$</td>
<td>$n$</td>
<td></td>
</tr>
<tr>
<td>Penalty function</td>
<td>approximate $I$ $I$ $x(k_n,t), u(k_n,t)$</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>Dynamic programming</td>
<td>exact ODE PDE$^c$ $I$ $u(t)$</td>
<td>$n$</td>
<td></td>
</tr>
<tr>
<td>Neighboring extremal</td>
<td>exact ODE ODE $\mathcal{H}_u = 0$ $\lambda(t_0)$</td>
<td>$2n$</td>
<td></td>
</tr>
<tr>
<td>Quasilinearization</td>
<td>exact $I$ $I$ $\mathcal{H}_u = 0$ $x(t), \lambda(t)$</td>
<td>$2n^d$</td>
<td></td>
</tr>
<tr>
<td>Gradient</td>
<td>exact ODE ODE $I$ $u(t)$</td>
<td>$2n$</td>
<td></td>
</tr>
</tbody>
</table>

$^a$ODE: ordinary differential equation.

$^b$Iteration.

$^c$PDE: Partial differential equation; HJB equation; one dependent variable ($V$), ($n + 1$) independent variables ($x, t$), $\partial V/\partial x$ corresponds to $\lambda^T$.

$^d$Perturbation equation for $\Delta x(t)$ and $\Delta \lambda(t)$. 
Parametric Optimization

\[
\min_{u(t)} J = \phi[x(t_f)] + \int_{t_0}^{t_f} L[x(t),u(t)]\,dt
\]
subject to
\[
\dot{x}(t) = f[x(t),u(t)], \quad x(t_0) \text{ given}
\]

Control specified by a parameter vector, \( k \)
No adjoint equations

\[
u(t) = k
\]
\[
u(t) = k_0 + k_1 t + k_2 t^2 + \cdots, \quad k = \begin{bmatrix} k_0 & k_1 & \cdots & k_m \end{bmatrix}
\]
\[
u(t) = k_0 + k_1 t + k_2 t^2 + \cdots, \quad k = \begin{bmatrix} k_0 & k_1 & \cdots & k_m \end{bmatrix}
\]
\[
u(t) = k_0 + k_1 \sin \left( \frac{\pi t}{t_f - t_0} \right) + k_2 \cos \left( \frac{\pi t}{t_f - t_0} \right), \quad k = \begin{bmatrix} k_0 & k_1 & k_2 \end{bmatrix}
\]

Examples

Parametric Optimization

\[
\min_{u(t)} J = \phi[x(t_f)] + \int_{t_0}^{t_f} L[x(t),u(t)]\,dt
\]
subject to
\[
\dot{x}(t) = f[x(t),u(t)], \quad x(t_0) \text{ given}
\]

- Necessary and sufficient conditions for a minimum
- Use static search algorithm to find minimizing control parameter, \( k \)

\[
\frac{\partial J}{\partial k} = 0
\]
\[
\frac{\partial^2 J}{\partial k^2} > 0
\]
Parametric Optimization Example

\[
\min_{u(t)} J = \phi\left[ x(t_f) \right] + \int_{t_o}^{t_f} L[x(t), u(t)] \, dt
\]
subject to
\[
\dot{x}(t) = f[x(t), u(t)], \quad x(t_o) \text{ given}
\]

\[
u(t) \triangleq \left( k_0 + k_1 t + k_2 t^2 \right)
\]

\[
\dot{V} = \left( T_{\text{max}} - C_D \frac{1}{2} \rho V^2 S \right) / m - g \sin \gamma
\]

\[
\dot{\gamma} = \frac{1}{V} \left( \left[ C_L \left( k_0 + k_1 t + k_2 t^2 \right) \frac{1}{2} \rho V^2 S \right] / m - g \cos \gamma \right)
\]

\[
\dot{h} = V \sin \gamma
\]
\[
\dot{r} = V \cos \gamma
\]
\[
\dot{m} = -\left( SFC \right)(T)
\]

Legendre Polynomials

Solutions to Legendre’s differential equation
Legendre Polynomials

**Legendre Polynomials**

*Polynomials can be generated by Rodrigues’s formula*

\[ P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left[(x^2-1)^n\right] \]

**Optimizing Control:**

Find minimizing values of \( k_n \)

\[ u(x) = k_0 P_0(x) + k_1 P_1(x) + k_2 P_2(x) + k_3 P_3(x) + k_4 P_4(x) + k_5 P_5(x) + \cdots \]

\[ x = \frac{t}{t_f - t_0} \]

\[ P_0(x) = 1 \]
\[ P_1(x) = x \]
\[ P_2(x) = \frac{1}{2} (3x^2 - 1) \]
\[ P_3(x) = \frac{1}{2} (5x^3 - 3x) \]
\[ P_4(x) = \frac{1}{8} (35x^4 - 30x^2 + 3) \]
\[ P_5(x) = \frac{1}{8} (63x^5 - 70x^3 + 15x) \]

Control History Optimized with Legendre Polynomials Could be expressed as a Simple Power Series

\[ u^*(x) = k_0^* P_0(x) + k_1^* P_1(x) + k_2^* P_2(x) + k_3^* P_3(x) + k_4^* P_4(x) + k_5^* P_5(x) + \cdots \]

\[ u^*(x) = a_0^* + a_1^* x + a_2^* x^2 + a_3^* x^3 + a_4^* x^4 + a_5^* x^5 + \cdots \]

\[ a_0^* = k_0^* - k_2^* \left( \frac{1}{2} \right) + k_4^* \left( \frac{3}{8} \right) + \cdots \]
\[ a_1^* = k_1^* - k_3^* \left( \frac{3}{2} \right) + k_5^* \left( \frac{15}{8} \right) + \cdots \]
\[ a_2^* = k_2^* \left( \frac{3}{2} \right) - k_4^* \left( \frac{30}{8} \right) + \cdots \]
\[ \cdots \]
**Parametric Optimization: Collocation**

- Admissible controls occur at discrete times, \( k \)
- Cost and dynamic constraint are discretized
- “Pseudospectral” Optimal Control
  State and adjoint points may be connected by basis functions, e.g., Legendre polynomials
- Continuous solution approached as time interval decreased

\[
\min_{u_k} J = \phi \left[ x_{f_k} \right] + \sum_{k=0}^{k_f-1} L \left[ x_k, u_k \right] \\
\text{subject to} \\
x_{k+1} = f_k \left[ x_k, u_k \right], \quad x_0 \text{ given}
\]

http://en.wikipedia.org/wiki/Collocation_method
http://en.wikipedia.org/wiki/Legendre_polynomials
http://en.wikipedia.org/wiki/Pseudospectral_optimal_control

**Penalty Function Method**

**Balakrishnan’s “Epsilon” Technique**

- No integration of the dynamic equation
- Parametric optimization of the state and control history

\[
x(t) = x(k_x,t) \\
u(t) = u(k_u,t)
\]

\( \dim(k_x) \geq n \) \( \dim(k_u) \geq m \)

- Augment the integral cost function by the dynamic equation error

\[
\min_{x(t),u(t)} J = \phi \left[ x(t_f), t_f \right] + \int_{t_0}^{t_f} \left\{ L \left[ x(t), u(t), t \right] + \left( \frac{1}{\epsilon} \right) \left[ \left[ f \left[ x(t), u(t), t \right] - \dot{x}(t) \right]^T \{ \bullet \} \right] \right\} dt
\]

\( 1/\epsilon \) is the penalty for not satisfying the dynamic constraint
Penalty Function Method

- Choose reasonable starting values of state and control parameters
  - e.g., state and control satisfy boundary conditions
- Evaluate cost function

\[ J_0 = \phi[x_0(t_f)] + \int_{t_0}^{t_f} \left[ L[x(t), u(t)] + \left( \frac{1}{\varepsilon} \right) \left( f[x(t), u(t)] - \dot{x}_0(t) \right)^T \right] dt \]

Update state and control parameters (e.g., steepest descent)

\[
\begin{align*}
k_{x_i} & = k_{x_i} - \alpha \left[ \frac{\partial J}{\partial k_{x_i}} \right] \\
k_{u_i} & = k_{u_i} - \alpha \left[ \frac{\partial J}{\partial k_{u_i}} \right] \\
x_{i+1}(t) & = x(k_{x_i}, t) \\
u_{i+1}(t) & = u(k_{u_i}, t)
\end{align*}
\]

Re-evaluate cost with higher penalty
Repeat to convergence

\[ J_i \rightarrow J_{i+1} \rightarrow J^*, \quad \varepsilon \rightarrow 0, \quad f[x(t), u(t)] \rightarrow \dot{x}(t) \]

Neighboring Extremal Method

"Shooting Method": Integrate both state and adjoint vector forward in time

\[
\begin{align*}
\dot{x}_{k+1}(t) & = f[x_{k+1}(t), u_{k}(t)], \\
x_{0}(t_0) & \text{ given, initial guess for } u_{0}(t) \\
\dot{\lambda}_{k+1}(t) & = - \left[ \frac{\partial L}{\partial x_k}(t) + \lambda_{k+1}^T(t) F_k(t) \right]^T, \quad \lambda_{k}(t_0) \text{ given}
\end{align*}
\]

with

\[
u_{k+1}(t) \text{ defined by } \frac{\partial H[x_{k}(t), u_{k}(t), \lambda_{k+1}(t), t]}{\partial u} = \left[ L_{u_k}(t) + \lambda_{k+1}^T(t) G_k(t) \right] = 0
\]

... but how do you know the initial value of the adjoint vector?
Neighboring Extremal Method

All trajectories are optimal (i.e., “extremals”) for some cost function because

\[
\frac{\partial H}{\partial u} = H_u = \begin{bmatrix} L_u + \lambda^T G \end{bmatrix} = 0
\]

Integrating state equation computes a value for \( \phi[x(t_f)] \)

\[
x(t_f) = x(t_0) + \int_{t_0}^{t_f} f[x_{k+1}(t), u_k(t)]; \quad \phi[x(t_f)] \rightarrow \frac{\partial \phi[x(t_f)]}{\partial x} = \lambda^T(t_f)
\]

Use a learning rule to estimate the initial value of the adjoint vector, e.g.,

\[
\lambda_{k+1}^T(t_0) = \lambda_k^T(t_0) - \alpha \begin{bmatrix} \lambda_k^T(t_f) - \lambda_{desired} \end{bmatrix}
\]

Gradient-Based Methods
Gradient-Based Search Algorithms

Steepest Descent

\[ u_{k+1}(t) = u_k(t) - \varepsilon \left[ \frac{\partial H}{\partial u}(t) \right] \]

Newton Raphson

\[ u_{k+1}(t) = u_k(t) - \left[ \frac{\partial^2 H}{\partial u^2}(t) \right]^{-1} \left[ \frac{\partial H}{\partial u}(t) \right] \]

Generalized Direct Search

\[ u_{k+1}(t) = u_k(t) - K \left[ \frac{\partial H}{\partial u}(t) \right] \]
Numerical Optimization Using Steepest-Descent Algorithm

Iterative bidirectional procedure

\[ \dot{x}_k(t) = f[x_k(t), u_{k-1}(t)], \quad x(t_0) \text{ given} \]

*Use educated guess for } u_0(t) \text{ on first iteration*}

\[ \dot{\lambda}_k(t) = - \left[ \frac{\partial H}{\partial x} \right]_k^T = - \left[ L_x(t) + \lambda^T(t) F(t) \right]_k^T, \]

\[ \lambda(t_f) = \left\{ \frac{\partial \phi[x(t_f)]}{\partial x} \right\}^T \]

*Use } x_{k-1}(t) \text{ and } u_{k-1}(t) \text{ from previous step*}
Numerical Optimization Using Steepest-Descent Algorithm

\[
\left( \frac{\partial H}{\partial u} \right)_k = \left[ L_u(t) + \lambda^T(t)G(t) \right]_k \quad [E - L #3]
\]

\[
\begin{align*}
\mathbf{u}_{k+1}(t) &= \mathbf{u}_k(t) - \varepsilon \left[ \frac{\partial H}{\partial \mathbf{u}} \right]_{\mathbf{u}(t) = \mathbf{u}_k(t)}^T \\
&= \mathbf{u}_k(t) - \varepsilon \left[ L_u + \lambda^T(t)G(t) \right]_k^T
\end{align*}
\]

Use \( \mathbf{x}(t) \), \( \lambda(t) \), and \( \mathbf{u}(t) \) from previous step

Finding the Best Steepest-Descent Gain

\[
J_0 \left[ u_k(t) \right], \quad 0 < t < t_f : \text{Best solution from the previous iteration}
\]

Calculate the gradient, \( \frac{\partial H}{\partial u}(t) \), in \( 0 < t < t_f \)

\[
J_1 \left[ u_k(t) - \varepsilon \frac{\partial H}{\partial u}(t) \right], \quad 0 < t < t_f : \text{Steepest-descent calculation of cost (1)}
\]

\[
J_2 \left[ u_k(t) - 2\varepsilon \frac{\partial H}{\partial u}(t) \right], \quad 0 < t < t_f : \text{Steepest-descent calculation of cost (2)}
\]

\[
\begin{align*}
J(\varepsilon) &= a_0 + a_1\varepsilon + a_2\varepsilon^2 \\
\begin{bmatrix} J_0 \\
J_1 \\
J_2 \end{bmatrix} &= \begin{bmatrix} a_0 + a_1(0) + a_2(0)^2 \\

a_0 + a_1(\varepsilon_1) + a_2(\varepsilon_1)^2 \\
a_0 + a_1(2\varepsilon_1) + a_2(2\varepsilon_1)^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\
1 & (\varepsilon_1) & (\varepsilon_1)^2 \\
1 & (2\varepsilon_1) & (2\varepsilon_1)^2 \end{bmatrix} \begin{bmatrix} a_0 \\
a_1 \\
a_2 \end{bmatrix}
\end{align*}
\]

Solve for \( a_0 \), \( a_1 \), and \( a_2 \)

Find \( \varepsilon^* \) that minimizes \( J(\varepsilon) \)

\[
J_{\varepsilon^*} \left[ u_{\varepsilon^*}(t) = u_k(t) - \varepsilon^* \frac{\partial H}{\partial u}(t) \right], \quad 0 < t < t_f : \text{Best steepest-descent calculation of cost}
\]

Go to next iteration
Steepest-Descent Algorithm for Problem with Terminal Constraint

\[
\min_{u(t)} J = \phi \left[ x(t_f) \right] + \int_{t_i}^{t_f} L \left[ x(t), u(t) \right] \, dt
\]

\[
\psi \left[ x(t_f) \right] \equiv 0 \text{ (scalar)}
\]

\[
\frac{\partial H_C}{\partial u} = \left[ \frac{\partial H_0}{\partial u} - \left( \frac{a}{b} \right) \frac{\partial H_1}{\partial u} \right] = 0
\]

\text{Chose } u_{k+1}(t) \text{ such that}

\[
u_{k+1}(t) = u_k(t) - \varepsilon \frac{\partial H_C}{\partial u} \bigg|_{u(t)=u_k(t)}
\]

\[
u_{k+1}(t) = u_k(t) - \varepsilon \left[ L_u^T + G^T \lambda_k(t) \right] + \frac{1}{b} G^T \lambda_k(t) \psi_k \left[ x(t_f) \right]
\]

Zero Gradient Algorithm for Quadratic Control Cost

\[
\min_{u(t)} J = \phi \left[ x(t_f) \right] + \int_{t_i}^{t_f} \left\{ L \left[ x(t) \right] + \frac{1}{2} u^T(t) Ru(t) \right\} \, dt
\]

\[
H \left[ x(t), u(t), \lambda(t) \right] = \left\{ L \left[ x(t) \right] + \frac{1}{2} u^T(t) Ru(t) \right\} + \lambda^T(t) f \left[ x(t), u(t) \right]
\]

Optimality condition:

\[
\frac{\partial H}{\partial u}(t) = H_u(t) = \left[ u^T(t) R + \lambda^T(t) G(t) \right] \equiv 0
\]
Zero Gradient Algorithm for Quadratic Control Cost

\[ \frac{\partial H}{\partial u}(t) = H_\alpha(t) = \left[ u^T(t)R + \lambda^T(t)G(t) \right] = 0 \]

Optimal control, \( u^*(t) \)

\[
\begin{align*}
    u^*_k(t) &= -\lambda^*_k(t)G^*_k(t) \\
    u^*_k(t) &= -R^{-1}G^*_k(t)\lambda^*_k(t)
\end{align*}
\]

But \( G_k(t) \) and \( \lambda_k(t) \) are sub-optimal before convergence, and optimal control cannot be computed in single step

\[\therefore \text{Chose } u_{k+1}(t) \text{ such that} \]

\[
\begin{align*}
    u_{k+1}(t) &= (1 - \varepsilon)u_k(t) - \varepsilon\left[ R^{-1}G_k^T(t)\lambda_k(t) \right] \\
    \varepsilon &\triangleq \text{Relaxation parameter} < 1
\end{align*}
\]

Stopping Conditions for Numerical Optimization

- Computed total cost, \( J \), reaches a theoretical minimum, e.g., zero
  \[ J_{k+1} = 0 + \varepsilon \]
  \[ J_{k+1} > J_k - \varepsilon \]

- Convergence of \( J \) is essentially complete

- Control gradient, \( H_\alpha(t) \), is essentially zero throughout \([t_0, t_f]\)

- Terminal cost/constraint is satisfied, and integral cost is “good enough”
  \[ \phi_{k+1}(t_f) = 0 + \varepsilon, \text{ or } \psi_{k+1}(t_f) = 0 + \varepsilon, \text{ and } \int_{t_0}^{t_f} L[x(t), u(t)]dt < \delta \]
Optimal Treatment of an Infection

Model of Infection and Immune Response

- $x_1 =$ Concentration of a pathogen, which displays antigen
- $x_2 =$ Concentration of plasma cells, which are carriers and producers of antibodies
- $x_3 =$ Concentration of antibodies, which recognize antigen and kill pathogen
- $x_4 =$ Relative characteristic of a damaged organ [0 = healthy, 1 = dead]
Infection Dynamics

Fourth-order ordinary differential equation, including effects of therapy (control)

\[
\begin{align*}
\dot{x}_1 &= (a_{11} - a_{12}x_3)x_1 + b_1u_1 + w_1 \\
\dot{x}_2 &= a_{21}(x_4)a_{22}x_1x_3 - a_{23}(x_2 - x_2^*) + b_2u_2 + w_2 \\
\dot{x}_3 &= a_{31}x_2 - (a_{32} + a_{33}x_1)x_3 + b_3u_3 + w_3 \\
\dot{x}_4 &= a_{41}x_1 - a_{42}x_4 + b_4u_4 + w_4
\end{align*}
\]

Uncontrolled Response to Infection

![Graphs showing uncontrolled response to infection](image)
Cost Function to be Minimized by Optimal Therapy

\[ J = \frac{1}{2} \left( p_{11} x_1^2 + p_{44} x_4^2 \right) + \frac{1}{2} \int_{t_0}^{t_f} \left( q_{11} x_1^2 + q_{44} x_4^2 + r u^2 \right) dt \]

- Tradeoffs between final values, integral values over a fixed time interval, state, and control
- Cost function includes weighted square values of
  - Final concentration of the pathogen
  - Final health of the damaged organ (0 is good, 1 is bad)
  - Integral of pathogen concentration
  - Integral health of the damaged organ (0 is good, 1 is bad)
  - Integral of drug usage
- Drug cost may reflect physiological cost (side effects) or financial cost

Examples of Optimal Therapy

- \( u_1 \) = Pathogen killer
- \( u_2 \) = Plasma cell enhancer
- \( u_3 \) = Antibody enhancer
- \( u_4 \) = Organ health enhancer

Unit cost weights
Effects of Increased Drug “Cost”

$r = 100$

Next Time:
Minimum-Time and -Fuel Problems

Reading
OCE: Section 3.5, 3.6
Supplemental Material

Examples of Equality Constraints

\[ c[x(t), u(t)] \equiv 0 \]

Pitch Moment = 0 = fcn(Mach Number, Stabilator Trim Angle)

\[ c[u(t)] \equiv 0 \]

Stabilator Trim Angle – constant = 0

\[ c[x(t)] \equiv 0 \]

Altitude – constant = 0
Minimum-Error-Norm Solution

- Euclidean error norm for linear equation
  \[ \| Ax - y \|_2^2 = (Ax - y)^T (Ax - y) \]

- Necessary condition for minimum error
  \[ \frac{\partial}{\partial x} \| Ax - y \|_2^2 = 2 (Ax - y)^T = 0 \]

- Express \( x \) as right pseudoinverse
  \[
  2Ax - y = 2\left\{ A \left( A^T (AA^T)^{-1} \right)^T y \right\} = 2\left\{ (AA^T)(AA^T)^{-1} y - y \right\} \\
  = 2 \{ y - y \}^T = 0 
  \]

- Therefore, \( x \) is the minimizing solution, as long as \( AA^T \) is non-singular