Topics

- Necessary Conditions of Optimality
- Shooting Method
- Gradient Method
- Quasi-Linearization Method
**Optimal Control Problem**

- **Performance Index (to minimize / maximize):**
  \[ J = \varphi(t_f, X_f) + \int_{t_0}^{t_f} L(t, X, U) \, dt \]

- **Path Constraint:**
  \[ \dot{X} = f(t, X, U) \]

- **Boundary Conditions:**
  \[ X(0) = X_0 : \text{Specified} \]
  \[ t_f : \text{Fixed}, \quad X(t_f) : \text{Free} \]

**Necessary Conditions of Optimality:**

- **State Equation**
  \[ \dot{X} = \frac{\partial H}{\partial \lambda} = f(t, X, U) \]

- **Costate Equation**
  \[ \dot{\lambda} = -\left( \frac{\partial H}{\partial X} \right) \]

- **Optimal Control Equation**
  \[ \frac{\partial H}{\partial U} = 0 \]

- **Boundary Condition**
  \[ \lambda_f = \frac{\partial \varphi}{\partial X_f} \quad X(t_0) = X_0 : \text{Fixed} \]
Necessary Conditions of Optimality: Some Comments

- State and Costate equations are dynamic equations. **If one is stable, the other turns out to be unstable!**
- Optimal control equation is a stationary equation.
- Boundary conditions are split: it leads to **Two-Point-Boundary-Value Problem (TPBVP)**.
- State equation develops forward whereas Costate equation develops backwards.
- It is known as **“Curse of Complexity”** in optimal control.
- Traditionally, TPBVPs demand computationally-intensive iterative numerical procedures, which lead to “open-loop” control structure.
Shooting Method

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Necessary Conditions of Optimality

- State Equation
  \[ X = \frac{\partial H}{\partial \lambda} = f(t, X, U) \]

- Costate Equation
  \[ \dot{\lambda} = -\left( \frac{\partial H}{\partial X} \right) = g(t, X, \lambda, U) \]

- Optimal Control Equation
  \[ \left( \frac{\partial H}{\partial U} = 0 \right) \Rightarrow U = \psi(X, \lambda) \]

- Boundary Condition
  \[ \lambda_f = \frac{\partial \phi}{\partial X_f} \quad X(t_0) = X_0 : \text{Fixed} \]

**Guess**: \( \lambda_0 \)
Shooting Method Philosophy

- Guess the initial condition for costate
- Compute the control at each grid point
- Propagate the state and costate
- Calculate the final boundary condition and error in the costate at the final time
- Correct the costate vector at the initial time based on this error at the final time
- Repeat the procedure

Shooting Method

- Form a Meta State Vector $Z = \begin{bmatrix} X \\ \lambda \end{bmatrix}$. This implies $dZ = \begin{bmatrix} dX \\ d\lambda \end{bmatrix}$.

- Guess $\lambda(t_0)$. Note that $X(t_0)$ is given. This leads to

$$Z \equiv \begin{bmatrix} X \\ \lambda \end{bmatrix} = F(Z) \quad \text{(1)}$$

$$Z(t_0) = \begin{bmatrix} X(t_0) \\ \lambda(t_0) \end{bmatrix}$$

- Obtain the linearized Error Dynamics Equation

$$dZ = \frac{\partial F}{\partial Z} dZ \quad \text{(2)}$$
Shooting Method

- Define a State Transition Matrix (STM) $\Phi$, such that at any two times $t_i$ and $t_f$,

\[
\frac{dZ(t_i)}{dt} = \Phi(t_i, t_f) \frac{dZ(t_f)}{dt}
\]  

(3)

- The dynamics and initial conditions for the STM can be shown to be

\[
\dot{\chi} = \left[ \frac{dF}{dZ} \right] \Phi \\
\Phi(t_0, t_f) = I_{n \times n}
\]

(4)

- Numerically integrate the equations (2) and (4) from $t_0$ to $t_f$, solving for the optimal control $U$ at each instant of time.

Shooting Method

- Finally, at $t = t_f$,

\[
\frac{dZ_f}{dt} = \left[ \frac{dX_f}{dM} \right] = \Phi(t_f, t_0) \frac{dZ_0}{dt}
\]

(5)

- Thus, at $t = t_0$,

\[
\frac{dZ_0}{dt} = \left[ \frac{dX_0}{d\lambda_0} \right] = \Phi^{-1}(t_f, t_0) \frac{dZ_f}{dt}
\]

(6)

- Since $X_0$ is fixed, force $dX_0 = 0$. Update only $\lambda_0$. Repeat until convergence.
Shooting Method

- Computational Load Reduction

Partition the STM (Φ) as \( \Phi = \begin{bmatrix} \Phi_1 & \Phi_2 \end{bmatrix} \). Then,

\[
dZ_f = \Phi_{1f} \Delta x_0 + \Phi_{2f} \Delta \lambda_0 = \Phi_{2f} \Delta \lambda_0
\]

\( (7) \)

Shooting Method

- For convenience, let \( h = (\lambda_f)_{n+1} \) be the vector of \( n \) boundary conditions at \( t_f \). Then,

\[
\left( \frac{\partial h}{\partial Z} \right)_f dZ_f = dh = (\lambda_f - \lambda_f^*)_{n+1}
\]

\( (8) \)

Where, \( \lambda_f^* \) is the true (desired) value of \( \lambda_f \).

- Finally, at \( t = t_f \),

\[
\Delta \lambda_0 = \left( \left( \frac{\partial h}{\partial Z} \right)_f \Phi_{2f} \right)^{-1} dh
\]

\( (9) \)

- Hence, obtain \( \Delta \lambda_0(k) \) and update \( \lambda_0(k) \) to \( \lambda_0(k+1) \). Repeat until convergence.
**Problems in Shooting Method**

- Sensitivity of the procedure to the initial guess value of costate
- Costates do not have ‘physical meaning’: complicates the issue of selecting ‘good’ initial values *(it is usually done through guessing a control history)*
- Costate equation is normally unstable for stable state dynamics: Long-duration prediction is not good!

**Multiple Shooting Approach**

- Strategy: “Divide-and-Rule”; i.e. divide the control application duration to multiple segments and solve the individual segments independently (possibly in a parallel setting to speed up the solution).
- This approach is called “Multiple Shooting” method.
- It brings in additional constraints of continuity and smoothness at the ‘joining points’.
**Gradient Method**

**Prof. Radhakant Padhi**

Dept. of Aerospace Engineering  
Indian Institute of Science - Bangalore

---

**Gradient Method**

- **Assumptions:**
  - State equation satisfied
  - Costate equation satisfied
  - Boundary conditions satisfied

- **Strategy:**
  - Satisfy the optimal control equation
Gradient Method

\[ \delta \bar{J} = \left( \delta X \right)^T \begin{bmatrix} \frac{\partial \phi}{\partial X} & \lambda_f \end{bmatrix} \\
+ \int_{t_0}^{t_f} \left( (\delta X)^T \frac{\partial H}{\partial X} \right) dt \\
+ \int_{t_0}^{t_f} \left( (\delta U)^T \frac{\partial H}{\partial U} \right) dt \\
+ \int_{t_0}^{t_f} \left( \delta \lambda \right)^T \frac{\partial H}{\partial \lambda} dt \]

Gradient Method

- After satisfying the state & costate equations and boundary conditions, we have

\[ \delta \bar{J} = \int_{t_0}^{t_f} (\delta U)^T \left[ \frac{\partial H}{\partial U} \right] dt \]

- Select

\[ \delta U(t) = -\tau \left[ \frac{\partial H}{\partial U} \right], \quad \tau > 0 \]

- This leads to

\[ \delta \bar{J} = -\tau \int_{t_0}^{t_f} \left[ \frac{\partial H}{\partial U} \right]^T \left[ \frac{\partial H}{\partial U} \right] dt \]
Gradient Method

- We select

\[ \delta U^i (t) = \left[ U^{i+1} (t) - U^i (t) \right] = -\tau \left[ \frac{\partial H}{\partial U} \right] \]

- This lead to

\[ U^{i+1} (t) = U^i (t) - \tau \left[ \frac{\partial H}{\partial U} \right] \]

- Note:

\[ \delta J = -\tau \int_{t_0}^{t_f} \left[ \frac{\partial H}{\partial U} \right]^T \left[ \frac{\partial H}{\partial U} \right] dt \leq 0 \]

- Eventually,

\[ \delta J = 0 \quad \Rightarrow \quad \frac{\partial H}{\partial U} = 0 \]

Gradient Method: Procedure

- Assume a control history (not a trivial task)
- Integrate the state equation forward
- Integrate the costate equation backward
- Update the control solution
  - This can either be done at each step while integrating the costate equation backward or after the integration of the costate equation is complete
- Repeat the procedure until convergence

\[ \int_{t_0}^{t_f} \left[ \frac{\partial H}{\partial U} \right]^T \left[ \frac{\partial H}{\partial U} \right] dt \leq \gamma \quad \text{(a pre-selected constant)} \]
Gradient Method: Selection of $\tau$

- Select $\tau$ so that it leads to a certain percentage reduction of $J$
- Let the percentage be $\alpha$
- Then \[
\tau = \frac{\alpha}{100} \left[ \int_{t_0}^{t_f} \frac{\partial H}{\partial U} \frac{\partial H}{\partial U} dt \right]
\]
- This leads to

A Challenging Problem

Objective:
Air-to-air missiles are usually launched from an aircraft in the forward direction. However, the missile should turn around and intercept a target “behind the aircraft”.

To execute this task, the missile should turn around by $-180^\circ$ and lock onto its target (after that it can be guided by its own homing guidance logic).

Note: Every other case can be considered as a subset of this extreme scenario!
A Challenging Problem

MATHEMATICAL PERSPECTIVE:
• Minimum time optimization problem
• Fixed initial conditions and free final time problem

SYSTEM DYNAMICS:
Equations of motion for a missile in vertical plane. The non-dimensional equations of motion (point mass) in a vertical plane are:

\[ M' = -S_w M^2 C_D - \sin(\gamma) + T_w \cos(\alpha) \]
\[ \gamma' = \frac{1}{M} [S_w M^2 C_L + T_w \sin(\alpha) - \cos(\gamma)] \]

where prime denotes differentiation with respect to the non-dimensional time \( \tau \)

A Challenging Problem

The non-dimensional parameters are defined as follows:

\[ \tau = \frac{g}{at}; \quad T_w = \frac{T}{mg}; \quad S_w = \frac{\rho a^2 S}{2mg}; \quad M = \frac{V}{a} \]

where \( M \) = flight Mach number
\( \gamma \) = flight path angle \( T = \) thrust
\( m \) = mass of the missile \( S = \) reference aerodynamic area
\( V \) = speed of the missile \( C_L = \) lift coefficient
\( C_D = \) drag coefficient \( g = \) the acceleration due to gravity
\( a = \) the local speed of sound \( \rho = \) the atmospheric density
\( t = \) flight time after launch

NOTE: \( C_L, C_D \) are usually functions of \( \alpha & M \) (tabulated data)
A Challenging Problem

COST FUNCTION:
Mathematically the problem is posed as follows to find the time minimizing cost function:

\[ J = \frac{1}{2} q \left( M_f - 0.8 \right)^2 + \int_{0}^{t_f} dt \]

Constraints \( \gamma(0) = 0^\circ \), \( M(0) = \) initial Mach number
\( \gamma(t_f) = -180^\circ \) [Note: \( M(t_f) \approx 0.8 \)]

A Challenging Problem

Choosing \( \gamma \) as the independent variable the equations are reformulated as follows:

\[ \frac{dM}{d\gamma} = \left( -S_{in}M^2C_D - \sin(\gamma) + T_e \cos(\alpha) \right) M \]
\[ \frac{dt}{d\gamma} = \frac{a M}{S_{in}M^2C_L - \cos(\gamma) + T_e \sin(\alpha)} \]

and the transformed cost function is

\[ J = \frac{1}{2} q \left( M_f - 0.8 \right)^2 + \int_{0}^{t_f} dt = \frac{1}{2} q \left( M_f - 0.8 \right)^2 + \int_{0}^{\gamma} a M \left( S_{in}M^2C_L - \cos(\gamma) + T_e \sin(\alpha) \right) d\gamma \]

(A difficult minimum-time problem has been converted to a relatively easier fixed final-flight path angle problem (with constraint: \( M(\gamma_f) = 0.8 \)).)
Assignment

Solve the problem using gradient method. Assume \( M(0) = 0.5 \) and engagement height as 5 km. Next, generate the trajectories and tabulate the values of \( M_f \) for various \( q \) values.

Use the following system parameters:

\[
\begin{align*}
    m &= 240 \text{ kg} \\
    S &= 0.0707 \text{ m}^2 \\
    T &= 24,000 \text{ N} \\
    C_d &= 0.5 \\
    C_l &= 3.12
\end{align*}
\]

Use standard atmosphere chart for the atmospheric data.

Quasi-Linearization Method

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
Quasi-Linearization Method

Problem:
Differential Equation: \( \dot{Z} = F(Z, t) \), \( Z := [X^T \lambda^T]^T \)
Boundary condition: \( \left\{ C(t_i), Z(t_i) \right\} = C_i^T Z_i = b_i \)
\( t_i \in t, \quad i \in \{1, \ldots, n\} \)

Assumption:
This vector differential equation has a unique solution over \( t \in [t_0, t_f] \)

Trick:
The nonlinear multi-point boundary value problem is transformed into
a sequence of linear non-stationary boundary value problems, the solution
of which is made to approximate the solution of the true problem.

Quasi-Linearization Method

(1) Guess an approximate solution \( Z^N(t) \) \( (N = 1) \) (it need not satisfy the B.C.)

For updating this solution, proceed with the following steps:

(2) Linearize the system dynamics about \( Z^N(t) \)

\( \Delta Z^N = \frac{\partial F}{\partial Z} \Delta Z^N \), where, \( \Delta Z^N (t) \equiv Z^{N+1}(t) - Z^N(t) \)

\( \Delta Z^N = \Lambda(t) \Delta Z^N \)

(3) Enforce the boundary with respect to the updated solution \( Z^{N+1}(t) \)

\( \left\{ C(t_i), Z^{N+1}(t_i) \right\} = \left\{ C(t_i), Z^N(t_i) + \Delta Z^N(t_i) \right\} = b_i \)

\( \left\{ C(t_i), \Delta Z^N(t_i) \right\} = -\left\{ C(t_i), Z^N(t_i) \right\} + b_i \)

Philosophy: Solve this linear system and update the solution!
Quasi-Linearization Method: Solution by STM Approach

1. From the linearized system dynamics, we can write
   \[ \dot{Z}^{NL} = A(t) (Z^{NL} - Z^N) \]
   \[ = A(t)Z^{NL} + \left[ F(Z^N, t) - A(t)Z^N \right] \]

2. The solution \( Z^{NL}(t) \) to the above equation is given by
   \[ Z^{NL}(t) = \Phi^{NL}(t, t_0) Z^{NL}(t_0) + p^{NL}(t) \]

3. The solution for STM \( \Phi^{NL}(t, t_0) \) can be obtained from the fact that
   it satisfies the following differential equation and boundary conditions
   \[ \frac{\partial}{\partial t} [\Phi^{NL}(t, t_0)] = A(t) \Phi^{NL}(t, t_0) \]
   \[ \Phi^{NL}(t_0, t_0) = I \]

4. The particular solution \( p^{NL}(t) \) can be obtained by observing that
   it satisfies the following differential equation and boundary condition
   \[ \frac{\partial}{\partial t} \left[ \Phi^{NL}(t, t_0) Z^{NL}(t_0) \right] + \dot{p}^{NL}(t) = A(t) \left[ \Phi^{NL}(t, t_0) Z^{NL}(t_0) + p^{NL}(t) \right] + F(Z^N, t) - A(t)Z^N \]
   \[ \dot{p}^{NL}(t) = A(t) p^{NL}(t) + F(Z^N, t) - A(t)Z^N \]

5. The boundary condition \( p^{NL}(t_0) \) can be obtained by observing that
   \[ Z^{NL}(t_0) = \Phi^{NL}(t_0, t_0) Z^{NL}(t_0) + p^{NL}(t_0) \]
   \[ p^{NL}(t_0) = 0 \]
Quasi-Linearization Method: Solution by STM Approach

(6) The boundary condition \( Z^{N+1}(t_0) \) can be obtained as follows

\[
\begin{align*}
\langle C(t), Z^{N+1}(t_0) \rangle &= b_i \\
\langle C(t), \Phi^{N+1}(t, t_0) Z^{N+1}(t_0) + p^{N+1}(t) \rangle &= b_i \\
\langle C(t), \Phi^{N+1}(t, t_0) Z^{N+1}(t_0) \rangle &= -\langle C(t), p^{N+1}(t) \rangle + b_i
\end{align*}
\]

Solve the above system to obtain \( Z^{N+1}(t_0) \)

Once \( Z^{N+1}(t_0) \) is determined, the solution \( Z^{N+1}(t) \) is available from the
STM solution:

\[
Z^{N+1}(t) = \Phi^{N+1}(t, t_0) Z^{N+1}(t_0) + \sum_{k \in \text{Particular solution}} \Phi^{N+1}(t, t_k) Z^{N+1}(t_k)
\]

Quasi-Linearization Method: Convergence Property

Under the assumption that the problem admits a unique solution for \( t \in [t_0, t_f] \), it can be shown that the sequence of vectors \( \{Z^{N+1}(t)\} \) converge to the true solution.

Moreover, the process can be shown to have "quadratic convergence" in general i.e., it can be shown that \( \|Z^{N+1}(t) - Z^N(t)\| \leq k \|Z^N(t) - Z^{N+1}(t)\| \), where \( k \neq f(N) \).

Further more, for a large class of systems, it can be shown to have "monotone convergence" as well, i.e. there won't be any over-shooting in the convergence process.

Assignment

Problem: Minimize $J = \frac{1}{2} \int_0^1 (x^2 + u^2) \, dt$ for the system $\dot{x} = -x^2 + u$, $x(0) = 10$.

Solution:

Hamiltonian: $H = \frac{1}{2} (x^2 + u^2) + \lambda (-x^2 + u)$

1) State Equation: $\dot{x} = -x^2 + u$
2) Optimal Control Equation: $u + \lambda = 0 \Rightarrow u = -\lambda$
3) Costate Equation: $\dot{\lambda} = -(\partial H / \partial x) = -x + 2\lambda x$
4) Boundary Conditions: $x(0) = 10$, $\lambda(1) = (\partial \Phi / \partial x) = 0$

Substituting the expression for $u$ in the state equation, we can write

\[
\begin{align*}
x &= -x^2 - \lambda, & x(0) &= 10 \\
\lambda &= -x + 2\lambda x, & \lambda(1) &= 0
\end{align*}
\]

Task: Solve this problem using quasi-linearization method.

References on Numerical Methods in Optimal Control Design

Survey of Classical Methods


Thanks for the Attention....!!