Lecture – 26

Classical Numerical Methods to Solve Optimal Control Problems

Dr. Radhakant Padhi
Asst. Professor
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
Necessary Conditions of Optimality in Optimal Control

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

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

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

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

- State and Costate equations are dynamic equations
- State equation develops forward whereas Costate equation develops backwards
- Optimal control equation is a stationary equation
- The formulation leads to Two-Point-Boundary-Value Problems (TPBVPs), which demand computationally-intensive iterative numerical procedures to obtain the optimal control solution
Classical Methods to Solve TPBVPs

- Gradient Method
- Shooting Method
- Quasi-Linearization Method
Gradient Method

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

- Strategy:
  - Satisfy the optimal control equation
Gradient Method

\[ \delta \bar{J} = \left( \delta X_f \right)^T \left[ \frac{\partial \phi}{\partial X_f} - \lambda_f \right] \]

\[ + \int_{t_0}^{t_f} \left( \delta X \right)^T \left[ \frac{\partial H}{\partial X} + \dot{\lambda} \right] \, dt \]

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

\[ + \int_{t_0}^{t_f} \left( \delta \lambda \right)^T \left[ \frac{\partial H}{\partial \lambda} - \dot{X} \right] \, dt \]
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:
  \[ \delta \bar{J} = -\tau \int_{t_0}^{t_f} \left[ \frac{\partial H}{\partial U} \right]^T \left[ \frac{\partial H}{\partial U} \right] dt \leq 0 \]

- Eventually,
  \[ \delta \bar{J} = 0 \quad \Rightarrow \quad \frac{\partial H}{\partial U} = 0 \]
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 $\bar{J}$
- Let the percentage be $\alpha$
- Then
  \[ \tau \int_{t_0}^{t_f} \left[ \frac{\partial H}{\partial U} \right]^T \left[ \frac{\partial H}{\partial U} \right] dt = \frac{\alpha}{100} |\bar{J}| \]
- This leads to
  \[ \tau = \frac{\frac{\alpha}{100} |\bar{J}|}{\int_{t_0}^{t_f} \left[ \frac{\partial H}{\partial U} \right]^T \left[ \frac{\partial H}{\partial U} \right] dt} \]
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° 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 Real-Life 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 Real-Life 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
\( \tau \) = thrust
\( m \) = mass of the missile
\( \rho \) = mass of the missile
\( V \) = speed of the missile
\( S \) = reference aerodynamic area
\( C_L \) = lift coefficient
\( C_D \) = drag coefficient
\( a \) = the local speed of sound
\( g \) = the acceleration due to gravity
\( \rho \) = the atmospheric density
\( t \) = flight time after launch

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

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

\[ J = \int_{0}^{t_f} dt \]

Constraints \( \gamma(0) = 0^\circ \), \( M(0) = \text{initial Mach number} \)

\( \gamma(t_f) = -180^\circ \), \( M(t_f) = 0.8 \)
A Real-Life Challenging Problem

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

\[
\frac{dM}{d\gamma} = \frac{\left(-S_w M^2 C_D - \sin(\gamma) + T_w \cos(\alpha)\right) M}{S_w M^2 C_L - \cos(\gamma) + T_w \sin(\alpha)}
\]

\[
\frac{dt}{d\gamma} = \frac{a M}{g\left(S_w M^2 C_L - \cos(\gamma) + T_w \sin(\alpha)\right)}
\]

and the transformed cost function is

\[
J = \int_0^{t_f} dt = \int_0^{-\pi} \frac{dt}{d\gamma} d\gamma = \int_0^{-\pi} \frac{a M}{g\left(S_w M^2 C_L - \cos(\gamma) + T_w \sin(\alpha)\right)} d\gamma
\]

(A difficult minimum-time problem has been converted to a relatively easier fixed final-time problem (with hard constraint: $M(\gamma_f) = 0.8$)!)
Task

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 (typical for an air-to-air missile):

- $m = 240 \text{ kg}$
- $S = 0.0707 \text{ m}^2$
- $T = 24,000 \text{ N}$
- $C_D = 0.5$
- $C_L = 3.12$

Use standard atmosphere chart for the atmospheric data.
Shooting Method

Dr. Radhakant Padhi
Asst. Professor
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
Necessary Conditions of Optimality (TPBVP): A Summary

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

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

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

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

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

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

- Obtain the linearized Error Dynamics Equation

$$
\ddot{Z} = \left[ \frac{\partial F}{\partial Z} \right] dZ \tag{2}
$$
Shooting Method

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

\[
dZ(t_j) = \Phi(t_j, t_i) \ dZ(t_i)
\]  
(3)

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

\[
\dot{\Phi} = \left[ \frac{\partial F}{\partial Z} \right] \Phi
\]  
(4)

\[
\Phi(t_0, t_0) = I_{2n \times 2n}
\]

- 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 $t = t_f$,

$$dZ_f = \begin{bmatrix} \frac{dX_f}{d\lambda_f} \end{bmatrix} = \Phi(t_f, t_0) \ dZ_0$$  \hfill (5)

- Thus, at $t = t_0$,

$$dZ_0 = \begin{bmatrix} \frac{dX_0}{d\lambda_0} \end{bmatrix} = \Phi^{-1}(t_f, t_0) \ dZ_f$$  \hfill (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 ($\Phi$) as $\Phi = \begin{bmatrix} \Phi_1 & \Phi_2 \end{bmatrix}$. Then,

$$dZ_f = \Phi_{1f} \, dX_0 + \Phi_{2f} \, d\lambda_0 = \Phi_{2f} \, d\lambda_0$$  \hspace{1cm} (7)
Shooting Method

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

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

(8)

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

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

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

(9)

- Hence, obtain \( d\lambda_0(k) \) and update \( \lambda_0(k) \) to \( \lambda_0(k + 1) \). Repeat until convergence.
Quasi-Linearization Method

Dr. Radhakant Padhi
Asst. Professor
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
Quasi-Linearization Method

Problem:

Differential Equation: \[ \dot{Z} = F(Z, t), \quad Z \triangleq [X^T \quad \lambda^T]^T \]

Boundary condition: \[ \langle C(t_i), Z(t_i) \rangle = C_i^T Z_i = b_i \]
\[ t_i \in t, \quad i \in \{1, \ldots, n\} \]

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 \( Z^N(t) \)

\[
\Delta \dot{Z}^N = \left[ \frac{\partial F}{\partial Z} \right]_{Z^N} \Delta Z^N, \quad \text{where, } \Delta Z^N(t) \triangleq Z^{N+1}(t) - Z^N(t) \text{ (To be found)}
\]

\[
\Delta \dot{Z}^N = A(t) \Delta Z^N
\]

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

\[
\langle C(t_i), Z^{N+1}(t_i) \rangle = \langle C(t_i), Z^N(t_i) + \Delta Z^N(t_i) \rangle = b_i
\]

\[
\langle C(t_i), \Delta Z^N(t_i) \rangle = -\langle C(t_i), Z^N(t_i) \rangle + b_i \quad \text{Philosophy: Solve this linear system and update the solution!}
\]
Quasi-Linearization Method: Solution by STM Approach

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

(2) The solution \( Z^{N+1}(t) \) to the above equation is given by
\[ Z^{N+1}(t) = \Phi^{N+1}(t, t_0) Z^{N+1}(t_0) + p^{N+1}(t) \]

(3) The solution for STM \( \Phi^{N+1}(t, t_0) \) can be obtained from the fact that it satisfies the following differential equation and boundary conditions
\[ \frac{\partial}{\partial t} \left[ \Phi^{N+1}(t, t_0) \right] = A(t) \Phi^{N+1}(t, t_0) \]
\[ \Phi^{N+1}(t_0, t_0) = I \]
Quasi-Linearization Method: Solution by STM Approach

(4) The particular solution \( p^{N+1}(t) \) can be obtained by observing that it satisfies the following differential equation and boundary condition:

Substituting the complete solution \( Z^{N+1}(t) \) in the original equation:

\[
\frac{\partial}{\partial t} \left[ \Phi^{N+1}(t, t_0) Z^{N+1}(t_0) \right] + \dot{p}^{N+1}(t) = A(t) \left[ \Phi^{N+1}(t, t_0) Z^{N+1}(t_0) + p^{N+1}(t) \right]
+ \left[ F(Z^N, t) - A(t) Z^N \right]
\]

\[
\dot{p}^{N+1}(t) = A(t) p^{N+1}(t) + \left[ F(Z^N, t) - A(t) Z^N \right]
\]

(5) The boundary condition \( p^{N+1}(t_0) \) can be obtained by observing that:

\[
Z^{N+1}(t_0) = \Phi^{N+1}(t_0, t_0) Z^{N+1}(t_0) + p^{N+1}(t_0)
\]

\[
p^{N+1}(t_0) = 0
\]
Quasi-Linearization Method: Solution by STM Approach

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

\[
\left\langle C(t_i), \ Z^{N+1}(t_i) \right\rangle = b_i \\
\left\langle C(t_i), \ \Phi^{N+1}(t_i, t_0)Z^{N+1}(t_0) + p^{N+1}(t_i) \right\rangle = b_i \\
\left\langle C(t_i), \ \Phi^{N+1}(t_i, t_0)Z^{N+1}(t_0) \right\rangle = -\left\langle C(t_i), \ p^{N+1}(t_i) \right\rangle + b_i
\]

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:

\[
Z^{N+1}(t) = \underbrace{\Phi^{N+1}(t, t_0)Z^{N+1}(t_0)}_{\text{STM}} + \underbrace{p^{N+1}(t)}_{\text{Particular solution}}
\]
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

\[
\left\|Z^{N+1}(t) - Z^N(t)\right\| \leq k \left\|Z^N(t) - Z^{N-1}(t)\right\|,
\]

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.

**A Demonstrative Example**

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

**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: \( u + \lambda = 0 \quad \Rightarrow \quad u = -\lambda \)

3) Costate Equation: \( \dot{\lambda} = -\left( \frac{\partial H}{\partial x} \right) = -x + 2\lambda x \)

4) Boundary Conditions: \( x(0) = 10, \quad \lambda (1) = \left( \frac{\partial \Phi}{\partial x} \right) = 0 \)

Substituting the expression for \( u \) in the state equation, we can write

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

**Task:** Solve this problem using shooting and quasi-linearization methods.
References on Numerical Methods in Optimal Control Design

Survey of Classical Methods


Thanks for the Attention...!