Topic: Transcription Method to Solve Optimal Control Problems

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

Topics

- Motivation
- Philosophy of Transcription Method
- Pseudospectral Transcription
- A Toy Problem
- An Application Problem
**Motivation**

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

---

**Optimal Control Formulation**

- Indirect Approach: - Variational Calculus
- Direct Approach:  
  - Dynamic Programming  
  - Transcription Method
Necessary Conditions of Optimality through Variational Calculus (Dualization)

- State Equation: \( \dot{X} = \frac{\partial H}{\partial \dot{X}} = f(t, X, U) \)
- Costate Equation: \( \dot{\lambda} = -\left( \frac{\partial H}{\partial X} \right) = g(t, X, U) \)
- Optimal Control Equation: \( \frac{\partial H}{\partial U} = 0 \) \( \Rightarrow \) \( U = \psi(X, \lambda) \)
- Boundary Condition: \( \lambda_f = \frac{\partial \phi}{\partial X} \) \( X(t_f) = X_0 : \text{Fixed} \)

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
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”
- It brings in additional constraints of continuity and smoothness at the ‘joining points’.
- Extension of this philosophy leads to “Transcription Method”: A Direct Approach!
Transcription Method

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

Philosophy of Transcription Method

- Convert the dynamic system variables into a finite set of static variables (or parameters)
- Pose an equivalent “static optimization” problem
- Solve this static optimization problem using static (paramter) optimization methods [e.g. using Nonlinear Programming (NLP)]
- Assess the accuracy
- Repeat the steps if necessary
Problem Objective & Philosophy

Objective:

\[ \text{Minimize} \quad J(x(.),u(.)) = E(x(t_0), x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t)) \, dt \]

Subject to

\[ \dot{x}(t) = f(x(t), u(t)) \]

with end point conditions

\[ e(x(t_0), x(t_f)) = 0 \]

and path constraints

\[ h(x(t), u(t)) \leq 0 \]

Philosophy: Select grid points, Discretize the states and control variables, Convert the problem to a nonlinear programming (NLP) problem and solve that problem, preferably in a computationally efficient manner.

---

Key Components of Direct Transcription

1. Choose discretization points (grid);
2. Approximate the trajectory \( x(t) \);
3. Discretize the state equation (derivative approximation);
4. Approximate the integration in the cost function.

**Free variables:**

\[ \left[ (x_{i_0}, u_{i_0}), (x_{i_1}, u_{i_1}), \ldots, (x_{i_N}, u_{i_N}) \right] \]

**Euler’s Method:**

\[ t_{k+1} = t_k + h \]

Approximate \( x(t) \) by piecewise linear function.

Ref: I. M. Ross, Lecture notes, Short course in AIAA GNC-2010, Toronto
Key Components of Direct Transcription

Reference: I. M. Ross, Lecture notes, Short course in AIAA GNC-2010, Toronto

Approximate the derivatives as: $x(t_k) \approx \frac{x(t_{k+1}) - x(t_k)}{h}$

In matrix form,

$$
\begin{pmatrix}
-1 & 1 & 0 & \cdots & 0 \\
1 & -1 & 1 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & -1 & 1
\end{pmatrix}
\begin{pmatrix}
x_0 \\
x_1 \\
\vdots \\
x_N
\end{pmatrix}
\approx
\begin{pmatrix}
f(x_0, u_0) \\
f(x_1, u_1) \\
\vdots \\
f(x_N, u_N)
\end{pmatrix}
$$

Approximate the integration as:

$$
\int_{t_0}^{t_f} F(t) \, dt \approx \sum_{i=0}^{N-1} h F(t_i)
$$

$$
= \frac{h}{2} \left[ F(x_0, u_0) + 2F(x_1, u_1) + \cdots + 2F(x_{N-1}, u_{N-1}) + F(x_N, u_N) \right]
$$

Direct Transcription

End point conditions

$$
x(t_0) = x_0^* \quad \Rightarrow \quad x_0 = x_0^*
$$

$$
\varphi(x(t_f)) = 0 \quad \Rightarrow \quad \varphi(x_N) = 0
$$

Path constraints

$$
h(x(t), u(t)) \leq 0 \quad \Rightarrow \quad
\begin{bmatrix}
h(x_0, u_0) \leq 0 \\
h(x_1, u_1) \leq 0 \\
\vdots \\
h(x_N, u_N) \leq 0
\end{bmatrix}
$$
**Direct Transcription: Other Ideas**

- Better Approximation of State Dynamics
  - Higher-order Finite difference
  - RK Methods
  - Polynomial Approximations in Segments
- Better Approximation of Cost Function
  - Higher-order Approximations
  - Quadrature Approximations
- Finite Element Approach

**Transcription Method**

- Accuracy
  - Higher number of grid points
  - Indirect Transcription

- Computational Efficiency
  - Sparse Algebra
  - Mesh Refinement
References


Pseudospectral Transcription

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

\[ \text{Minimize} \quad J = E(x(t_0), x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t)) \, dt \]

Subject to \( \dot{x}(t) = f(x(t), u(t)) \)

with end point conditions

\[ e(x(t_0), x(t_f)) = 0 \]

and path constraints

\[ h(x(t), u(t)) \leq 0 \]

**Philosophy:** Discretize the states (and the control) using Pseudospectral (PS) method, Convert the problem to a “lower-dimensional” nonlinear programming (NLP) problem and solve that problem in a computationally efficient manner.

Steps involved…

- **1. Approximate** \( x(t) \) and/or \( u(t) \)?
  \[ \dot{x}(t) = \sum_{n=0}^{N} a_n \phi_n(t) \quad \dot{u}(t) = \sum_{n=0}^{N} b_n \phi_n(t) \]

- **2. Selection of grid points**
  - How are these points selected?
  - Uniform grid is not a very good choice!

- **3. Discretize the differential equation using PS method**
  - Finite-difference Vs Spectral
  - Sparse Vs Dense differentiation matrix

- **4. Approximate the integral equation**
  - Quadrature rules

- **5. Apply an efficient finite optimization technique and solve the lower dimensional NLP problem.**
1. Approximation

It can be thought of as a function $\hat{x}$ which satisfies the boundary condition and makes the residual small:

$$\|\hat{x} - f\|_{SMALL}$$

$$\hat{x}(t) = f(x(t), u(t))$$

- Define Trial functions: $P_N := \{\phi_0, \phi_1, \ldots, \phi_N\}$

Approximate $\dot{x}(t) = \sum_{n=0}^{N} a_n \phi_n(t)$, $\dot{u}(t) = \sum_{n=0}^{N} b_n \phi_n(t)$

- Define Test functions: $(\chi_0, \chi_1, \ldots, \chi_N)$

$$\langle \chi_n, R \rangle = 0 \quad \forall n \in \{0,1,\ldots,N\}$$

2. Selection of grids

Ref: I. M. Ross, Lecture notes, Short course in AIAA GNC-2010, Toronto

Runge’s famous counter example disproving presumed convergence of interpolating polynomials over a uniform grid.
2. Selection of grids

Grid of collocation points (or grid points) \( t_n, n=0, \ldots, N \) are points such that it satisfies the state equation exactly at these points.

<table>
<thead>
<tr>
<th>Grid of collocation points</th>
<th>Increasing Generality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gauss-Lobatto</td>
<td>Yes</td>
</tr>
<tr>
<td>Gauss-Radau</td>
<td>Yes</td>
</tr>
<tr>
<td>Gauss</td>
<td>No</td>
</tr>
<tr>
<td>Uniform</td>
<td>No</td>
</tr>
</tbody>
</table>

3. Approximating the differential equation

Approximations: \( \hat{x}(t) = \sum_{n=0}^{N} a_{n} \phi_{n}(t), \quad \hat{u}(t) = \sum_{n=0}^{N} b_{n} \phi_{n}(t) \)

State Equation Constraint:
\[
\begin{align*}
\dot{x} &= f(x(t), u(t)) \\
\dot{\hat{x}} &= f(\hat{x}(t), \hat{u}(t)) \\
\sum_{n=0}^{N} \phi_{n}(t) a_{n} &= f\left(\sum_{n=0}^{N} a_{n} \phi_{n}(t), \sum_{n=0}^{N} b_{n} \phi_{n}(t)\right) \\
\end{align*}
\]

Multiply with \( \delta(t - t_n) \) on both sides:
\[
\sum_{n=0}^{N} \phi_{n}(t_n) a_{n} = f\left(\sum_{n=0}^{N} a_{n} \phi_{n}(t_n), \sum_{n=0}^{N} b_{n} \phi_{n}(t_n)\right), \quad n = 0, 1, \ldots, N
\]
3. Approximating the differential equation (Example)

- The Chebyshev polynomials
  \[ T_0(t) = 1 \]
  \[ T_1(t) = 2t - 1 \]
  \[ T_2(t) = 4t^2 - 8t + 1 \]
  \[ T_3(t) = 8t^3 - 48t^2 + 18t - 1 \]
  \[ T_4(t) = 32t^4 - 256t^3 + 640t^2 - 320t + 1 \]

- Polynomial Approximation
  \[ p(t) = a_0 T_0(t) + a_1 T_1(t) + a_2 T_2(t) + a_3 T_3(t) + a_4 T_4(t) \]

- The differential yields
  \[
  \begin{bmatrix}
  \dot{x}(t_1) \\
  \dot{x}(t_2) \\
  \dot{x}(t_3) \\
  \dot{x}(t_4)
  \end{bmatrix} =
  \begin{bmatrix}
  T_0(t_1) & T_0(t_2) & T_0(t_3) & T_0(t_4) \\
  T_1(t_1) & T_1(t_2) & T_1(t_3) & T_1(t_4) \\
  T_2(t_1) & T_2(t_2) & T_2(t_3) & T_2(t_4) \\
  T_3(t_1) & T_3(t_2) & T_3(t_3) & T_3(t_4)
  \end{bmatrix} \begin{bmatrix}
  a_0 \\
  a_1 \\
  a_2 \\
  a_3
  \end{bmatrix} +
  \begin{bmatrix}
  b_0 \\
  b_1 \\
  b_2 \\
  b_3
  \end{bmatrix}
  \]

- Select Grid Points & Evaluate: \( t_1, t_2, t_3, t_4 \)

---

4. Discretizing the integral equation - Gauss Quadratures

A quadrature rule is an approximation of the definite integral of a function, usually stated as a weighted sum of function values at specified points within the domain of integration.

\[
\int_{-1}^{1} L(x(t), u(t)) dt = \sum_{n=0}^{N} w_n L(x(t_n), \hat{u}(t_n))
\]

\[
J \equiv J^N = E(\hat{x}(t_N), \hat{u}(t_N)) + \frac{t_f - t_0}{2} \sum_{n=0}^{N} w_n L(\hat{x}(t_n), \hat{u}(t_n))
\]
Finally,

\[ \dot{x}(t) = \sum_{n=0}^{N} a_n \phi_n(t) \quad \dot{u}(t) = \sum_{n=0}^{N} b_n \phi_n(t) \]

Minimize,

\[ J^N = E(\dot{x}(t_0).\ddot{x}(t_N)) + \frac{t_f - t_0}{2} \sum_{n=0}^{N} w_n E(\dot{x}(t_n).\ddot{u}(t_n)) \]

Subject to,

\[ \sum_{n=0}^{N} \phi'_n(t_n) a_n = f(\dot{x}(t_n),\ddot{u}(t_n)) \quad 0 \leq n \leq N \]

with end point conditions,

\[ e(\dot{x}(t_0),\ddot{x}(t_N)) = 0 \]

and path constraints,

\[ h(\dot{x}(t_n),\ddot{u}(t_n)) \leq 0 \quad 0 \leq n \leq N \]

The optimal control problem has been simplified to a lower dimensional nonlinear programming problem.

---

A Toy Problem

Dr. Radhakant Padhi
Associate Professor
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
Toy Problem

Minimize \[ J = \int_0^1 u^*(t)dt \]

Subject to \[ \dot{x}(t) = x(t) + u(t) \]

Boundary condition \( (x(0), x(1)) = (1, e) \)

Subject to \[ |u(t)| \leq 1 \]

1. Decide on which polynomial we are planning as the trial function
   (i) Chebyshev Polynomials of the first kind. Say take N=4 (First 5 polynomials)
   \[ T_0(t) = 1, \quad T_1(t) = t, \quad T_2(t) = 2t^2 - 1 \quad T_3(t) = 4t^3 - 3t, \quad T_4(t) = 8t^4 - 8t^2 + 1 \]

(ii) Shifted Chebyshev polynomials for the interval [0,1]
    \[ T_0(t) = 1, \quad T_1(t) = 2t - 1 \quad T_2(t) = 8t^2 - 8t + 1 \quad T_3(t) = 32t^3 - 48t + 1 \quad T_4(t) = 128t^4 - 256t^2 + 160t^2 - 32t + 1 \]

Step 2. Define the collocation (grid) points

Shifted \[ t_i = \frac{1}{2} \frac{2a + b}{2} \frac{a - b}{N} \frac{(i \pi)}{N - 1} \quad i = 0, 1, \ldots, N - 1 \]
Toy Problem (Contd.)

Step 3. Approximate $x(t)$ and $u(t)$ using trial function

$$
\hat{x}(t) = \sum_{n=0}^{4} a_n T_n(t)
$$

$$
\hat{u}(t) = \sum_{n=0}^{4} b_n T_n(t)
$$

Step 4. Find the differentiation matrix and equate the state equation at the grid points

$$
\frac{d}{dt}(\hat{x}(t_i)) = \hat{x}(t_i) + \hat{u}(t_i)
$$

Toy Problem (Contd.)

Step 5. Compute $T_n(t)$ matrix for $t_i$ and equate the state equation, $i = 0,1,2,3,4$

Finally, we have

$$
T^{-1}(T \times D - I)
$$
Toy Problem (Contd.)

Step 6. Apply boundary conditions

\[
\begin{bmatrix}
1 \\
x(0) \\
x(1) \\
e
\end{bmatrix} =
\begin{bmatrix}
1 & -1 & 1 & -1 & 1 \\
1 & -1/\sqrt{3} & 0 & 1/\sqrt{3} & -1 \\
1 & 0 & -1 & 0 & 1 \\
1 & 1/\sqrt{3} & 0 & -1/\sqrt{3} & -1 \\
1 & 1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
a_3 \\
a_4
\end{bmatrix}
\]

This gives,

\[a_0 - a_1 + a_2 - a_3 + a_4 = 1\]
\[a_0 + a_1 + a_2 + a_3 + a_4 = 2.7182\]

Fix \(x(0) = 1\)

Fix \(x(1) = e\)

Toy Problem (Contd.)

Step 7. Approximate the integral equation

\[
J = \int_0^1 \hat{u}^2(t)\,dt \approx J^n = \frac{1}{2}\sum_{k=0}^n w_k \hat{u}^2(t_k)
\]

where

\[w_k = (b - a)\frac{n}{n+1} = \{1.57, 0.7854, 0.5236, 0.3927, 0.3142\}\]
Finally, we have

\[ J^N = \frac{1}{2} \sum_{k=0}^{N-1} w_k \hat{u}^2(t_k) \]
\[ \dot{x}(t_k) + \hat{u}(t_k) - \hat{x}(t_k) = 0 \]
\[ \hat{u}(t_k) - 1 \leq 0 \quad \text{for} \quad k = 0, 1, 2, 3, 4 \]

Define the augmented cost function,

\[ J^N = \frac{1}{2} \sum_{k=0}^{N-1} w_k \hat{u}^2(t_k) + \sum_{k=0}^{N-1} \lambda_k (\hat{x}(t_k) + \hat{u}(t_k) - \hat{x}(t_k)) + \sum_{k=0}^{N-1} \mu \lambda_k (\hat{u}(t_k) - 1) + \nu \lambda (\hat{x}(0) - 1) + \nu \lambda (\hat{x}(e) - 1) \]

Apply any KKT (or any other static optimization technique) for solving the optimal control problem.

### Toy Problem (Contd.)

**Define the augmented cost function**

\[ J^N = \frac{1}{2} \sum_{k=0}^{N-1} w_k \hat{u}^2(t_k) + \sum_{k=0}^{N-1} \lambda_k (\hat{x}(t_k) + \hat{u}(t_k) - \hat{x}(t_k)) + \sum_{k=0}^{N-1} \mu \lambda_k (\hat{u}(t_k) - 1) + \nu \lambda (\hat{x}(0) - 1) + \nu \lambda (\hat{x}(e) - 1) \]

Apply any KKT (or any other static optimization technique) for solving the optimal control problem.

### Toy Problem: Results

![Graph showing results](image)

\[ \sigma_0 = 1.7534 \quad \sigma_1 = 0.8503 \quad \sigma_2 = 0.1052 \quad \sigma_3 = 0.0088 \quad \sigma_4 = 0.0005 \]
**Toy Problem: Results**

![Graph showing control effort over time with time (sec) on the x-axis and control effort on the y-axis.](image)

- $b_0 = 0.0014$, $b_1 = 0.0004$, $b_2 = -0.0009$, $b_3 = 0.0006$, $b_4 = -0.0016$

---

**Minimum-time Front-to-back Turning of an Air-Launched Missile using Pseudo-Spectral Method**

Full Paper: Proceedings of 2012 IFAC EGNCA Workshop, Bangalore

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


### The problem

- Launch a missile from a aircraft in the forward direction and attack a target in the rear hemisphere.
- Missile has to reverse the flight path angle as quickly as possible so as to intercept the target.
- The problem is to reverse the flight path angle from an initial value (around 0°) to -180° in minimum time.
- After turning it should also have the required velocity (a constraint on the final Mach number) to intercept the target.
The problem –
Cost function and boundary conditions

Minimize
\[ J = \int_{0}^{t_f} dt \]
subject to the constraint
\[ \gamma(t_0) = \text{known (around 0°)}, \quad \gamma(t_f) = -180° \]
\[ M(t_0) = \text{known}, \quad M(t_f) = 0.8 \]

The problem is to reverse the flight path angle from 0° to -180° maintaining a final Mach number of 0.8 in minimum time.

System Dynamics

The point mass equations of motion:
\[ V(t) = \frac{D}{m} - \frac{W}{m} \sin \gamma + \frac{T}{m} \cos(\alpha + \delta) \]
\[ \dot{\gamma}(t) = \frac{1}{mV} (L - W \cos \gamma + T \sin(\alpha + \delta)) \]
\[ \dot{h}(t) = V \sin \gamma \]
\[ \dot{x}(t) = V \cos \gamma \]
System Dynamics

Ref: Han et al., “State constrained Agile Missile Control with Adaptive Critic Based Neural Networks”, JGCD, 2002

The non-dimensional parameters were considered as:

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

Non-dimensional point mass equations of motion:

\[
M'\tau = -S_w M^2 C_\alpha - \sin \gamma + T_w \cos(\alpha + \delta_w) \\
\gamma' = \frac{1}{M} \left( S_w M^2 C_\gamma - \cos \gamma + T_w \sin(\alpha + \delta_w) \right) \\
h' = \frac{\sigma x^2 M \sin \gamma}{g} \\
x' = \frac{\sigma x^2 M \cos \gamma}{g}
\]

Minimum time problem

- Free final time problem - state constrained problem, with bounds on control.

- Equations of motion are reformulated using flight path angle as the independent variable.

- Leads to a fixed final condition.

- Assumption: Flight path angle is a monotonic continuous function with respect to time.
\(y\) as independent variable

Modified state equations with respect to flight path angle

\[
\frac{dM}{d\gamma} = M' \gamma; \quad \frac{dt}{d\gamma} = \frac{1}{\gamma'}; \quad \frac{dh}{d\gamma} = h' \gamma; \quad \frac{dx}{d\gamma} = x' \gamma
\]

Modified cost function

\[
J = \int_{t_0}^{t_f} \frac{aM}{g(-S_w M'^2 C_s - \cos \gamma + T_w \sin(\alpha + \delta))} d\gamma
\]

The minimum time problem \(\rightarrow\) Hard constraint problem \(M(\gamma_f) = 0.8\)

Nonlinear programming problem

Minimize

\[
j' = \frac{t_f - t_0}{2} \sum_{k=0}^{N} w_k \left( \frac{a\dot{M}(\gamma_k)}{g(S_w \dot{\dot{M}}(\gamma_k))} \right)^2 C_s + T_w \sin(\dot{\alpha}(\gamma_k) + \delta(\delta_k)) - \cos \gamma_k
\]

Subject to

\[
\sum_{k=0}^{N} a_k t_k(\gamma_k) = \frac{-S_w \dot{\dot{M}}(\gamma_k)}{g(S_w \dot{\dot{M}}(\gamma_k))} C_s + T_w \cos(\dot{\alpha}(\gamma_k) + \delta(\delta_k)) - \sin \gamma_k
\]

Subject to path constraints

\[
|\dot{\gamma}(\gamma_k)| \leq 20^\circ \text{ and } |\ddot{\gamma}(\gamma_k)| \leq 72^\circ
\]

with end point conditions

\[
e_0(x_0) = \dot{M}(\gamma_0) = M_0, \text{ where } M_0 \in [0.3, 0.8]
\]

\[
e_f(x_f) = \dot{M}(\gamma_f) = 0.8
\]
Mach and AOA Vs Flight path angle
**Flight path angle history**

Effect of reducing the AOA
Effect of reducing the AOA on Mach number along with the flight path angle

Effect of the number of grids
## Selection of number of grids

<table>
<thead>
<tr>
<th>No. of grids</th>
<th>Computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.899679</td>
</tr>
<tr>
<td>10</td>
<td>1.022969</td>
</tr>
<tr>
<td>12</td>
<td>1.102450</td>
</tr>
<tr>
<td>20</td>
<td>1.652399</td>
</tr>
<tr>
<td>30</td>
<td>3.314869</td>
</tr>
</tbody>
</table>

Intel® Core™ 2 Quad CPU
Q6600 @2.40 GHz, 1.98GB RAM
Software: MATLAB 7.4

- The number of grids for analysis: 20
- 20 grids are comparable with 30

**Note:** Real-time implementation in “C” or “Assembly language” is expected to be much (at least 50 times) faster than the MATLAB Code

---

## Comparison of Chebyshev and Legendre approximation

Both lead to identical results!
Conclusions: Missile-turning Problem

- Real-hemisphere engagement is feasible (no need of “dog fight”!)
- Minimum-time flight path angle reversal is feasible with “realistic control force”
- Promising numerical results
  - Computationally very efficient & is a viable tool for optimal guidance
  - Chebyshev and Legendre approximations lead to identical results (serves as a verification)

References on Pseudo-spectral Methods for Optimal Control

Thanks for the Attention....!!

questions ... ??