AA203
Optimal and Learning-based Control

Direct methods for optimal control: fundamental concepts
Roadmap

Optimal control

- Open-loop
  - Indirect methods
  - Direct methods

- MPC

- Closed-loop
  - Adaptive optimal control
  - Model-free RL
  - Model-based RL

Linear methods
- DP
  - LQR
  - iLQR
  - DDP

Non-linear methods
- HJB / HJI
  - LQR
  - Reachability analysis

Reachability analysis
Agenda

• Introduction to direct methods
• Connection to indirect methods
• “State and control” and “control” parameterization methods

Readings: lecture notes and references therein, and also:

Optimal control problem

\[
\min \int_0^{t_f} g(x(t), u(t), t) \, dt
\]

\[
\dot{x}(t) = f(x(t), u(t), t), \quad t \in [0, t_f]
\]

\[
x(0) = x_0
\]

\[
x(t_f) \in M_f = \{ x \in \mathbb{R}^n : F(x) = 0 \}
\]

\[
u(t) \in U \subseteq \mathbb{R}^m, \quad t \in [0, t_f]
\]

For simplicity:
- We assume the terminal cost \( h \) is equal to 0
- We assume \( t_0 = 0 \)

• Direct Methods:
  1. Transcribe (OCP) into a nonlinear, constrained optimization problem
  2. Solve the optimization problem via nonlinear programming

• Indirect Methods:
  1. Apply necessary conditions for optimality to (OCP)
  2. Solve a two-point boundary value problem
Transcription into nonlinear programming

Forward Euler time discretization

1. Select a discretization $0 = t_0 < t_1 < \cdots < t_N = t_f$ for the interval $[0, t_f]$ and, for every $i = 0, \ldots, N - 1$, define $x_i \sim x(t), \ u_i \sim u(t), \ t \in [t_i, t_{i+1})$ and $x_0 \sim x(0)$

2. By denoting $h_i = t_{i+1} - t_i$, (OCP) is transcribed into the following nonlinear, constrained optimization problem

\[
\min_{(x_i, u_i)} \sum_{i=0}^{N-1} h_i g(x_i, u_i, t_i)
\]

\[
x_{i+1} = x_i + h_i f(x_i, u_i, t_i), \quad i = 0, \ldots, N - 1
\]

\[
u_i \in U, \quad i = 0, \ldots, N - 1, \quad F(x_N) = 0
\]
Connection to indirect methods (informal)

Simplified Formulation

\[ \min \int_0^{t_f} g(x(t), u(t)) \, dt \]
\[ \dot{x}(t) = f(x(t), u(t)), \ t \in [0, t_f] \]
\[ x(0) = x_0 \]

Related non-linear program (NLOP)

After discretization in time:

\[ \min_{(x_i, u_i)} \sum_{i=0}^{N-1} h_i g(x_i, u_i) \]  \hspace{1cm} (NLOP)
\[ x_i + h_i f(x_i, u_i) - x_{i+1} = 0, \quad i = 0, \ldots, N-1 \]

KKT conditions for (NLOP) converge to the necessary optimality conditions for (OCP), given by the Pontryagin’s Minimum Principle, when \( h_i \to 0 \)
Connection to indirect methods

KKT Related to (NLOP)

Denote the Lagrangian related to (NLOP) as

\[ L = \sum_{i=0}^{N-1} h_i g(x_i, u_i) + \sum_{i=0}^{N-1} \lambda'_i (x_i + h_i f(x_i, u_i) - x_{i+1}) \]

Then, the KKT conditions related to (NLOP) read as:

- Derivative w.r.t. \( x_i \): 
  \[ h_i \frac{\partial g}{\partial x_i}(x_i, u_i) + \lambda_i - \lambda_{i-1} + h_i \frac{\partial f}{\partial x_i}(x_i, u_i) \lambda_i = 0 \]

- Derivative w.r.t. \( u_i \): 
  \[ h_i \frac{\partial g}{\partial u_i}(x_i, u_i) + h_i \frac{\partial f}{\partial u_i}(x_i, u_i) \lambda_i = 0 \]

Related non-linear program (NLOP)

After discretization in time:

\[ \min_{(x_i, u_i)} \sum_{i=0}^{N-1} h_i g(x_i, u_i) \quad \text{(NLOP)} \]

\[ x_i + h_i f(x_i, u_i) - x_{i+1} = 0, \quad i = 0, \ldots, N - 1 \]
Connection to indirect methods

KKT Related to (NLOP)

Denote the Lagrangian related to (NLOP) as

\[ \mathcal{L} = \sum_{i=0}^{N-1} h_i g(x_i, u_i) + \sum_{i=0}^{N-1} \lambda_i' (x_i + h_i f(x_i, u_i) - x_{i+1}) \]

Then, the KKT conditions related to (NLOP) read as:

- Derivative w.r.t. \( x_i \):
  \[ h_i \frac{\partial g}{\partial x_i} (x_i, u_i) + \lambda_i - \lambda_{i-1} + h_i \frac{\partial f}{\partial x_i} (x_i, u_i)' \lambda_i = 0 \]

- Derivative w.r.t. \( u_i \):
  \[ h_i \frac{\partial g}{\partial u_i} (x_i, u_i) + h_i \frac{\partial f}{\partial u_i} (x_i, u_i)' \lambda_i = 0 \]

Back to the continuous-time formulation

We finally obtain:

\[ \frac{\lambda_i - \lambda_{i-1}}{h_i} = - \frac{\partial f}{\partial x_i} (x_i, u_i)' \lambda_i - \frac{\partial g}{\partial x_i} (x_i, u_i) \]

\[ \frac{\partial f}{\partial u_i} (x_i, u_i)' \lambda_i + \frac{\partial g}{\partial u_i} (x_i, u_i) = 0 \]

Let \( p(t) = \lambda_i \) for \( t \in [t_i, t_{i+1}) \), \( i = 0, ..., N - 1 \) and \( p(0) = \lambda_0 \). In the limit \( h_i \to 0 \), one “obtains” necessary conditions for (OCP):

\[ \dot{p}(t) = - \frac{\partial f}{\partial x} (x(t), u(t)') p(t) - \frac{\partial g}{\partial x} (x(t), u(t)) \]

\[ \frac{\partial f}{\partial u} (x(t), u(t)') p(t) + \frac{\partial g}{\partial u} (x(t), u(t)) = 0 \]
Pontryagin’s Minimum Principle

Simplified Formulation

\[
\min \int_0^{t_f} g(x(t), u(t)) \, dt
\]

\[
\dot{x}(t) = f(x(t), u(t)), \quad t \in [0, t_f]
\]

(OCP)

\[
x(0) = x_0
\]

Pontryagin’s Minimum Principle (PMP)

The necessary optimality conditions for (OCP) are given by the coupled differential equations

- Co-state equation:
  \[
  \dot{p}(t) = -\frac{\partial f}{\partial x}(x(t), u(t))' p(t) - \frac{\partial g}{\partial x}(x(t), u(t))
  \]

- Control equation:
  \[
  \frac{\partial f}{\partial u}(x(t), u(t))' p(t) + \frac{\partial g}{\partial u}(x(t), u(t)) = 0
  \]

- Dynamics:
  \[
  \dot{x}(t) = f(x(t), u(t))
  \]
Back to direct methods: solution approaches

1. state and control parameterization methods

2. control parameterization methods
Transcription into nonlinear programming
(state and control parametrization method)

\[
\min \int_0^{t_f} g(x(t), u(t), t) \, dt
\]

\(\dot{x}(t) = f(x(t), u(t), t), \quad t \in [0, t_f]\)  
(OCP)

\[x(0) = x_0, \quad x(t_f) \in M_f\]

\[u(t) \in U \subseteq \mathbb{R}^m, \quad t \in [0, t_f]\]

Forward Euler time discretization

1. Select a discretization \(0 = t_0 < t_1 < \cdots < t_N = t_f\) for the interval \([0, t_f]\) and, for every \(i = 0, \ldots, N - 1\), define \(x_i \sim x(t), \quad u_i \sim u(t), \quad t \in [t_i, t_{i+1}]\) and \(x_0 \sim x(0)\)

2. By denoting \(h_i = t_{i+1} - t_i\), (OCP) is transcribed into the following nonlinear, constrained optimization problem

\[
\min_{(x_i, u_i)} \sum_{i=0}^{N-1} h_i g(x_i, u_i, t_i)
\]

(NLOP)

\[x_{i+1} = x_i + h_i f(x_i, u_i, t_i), \quad i = 0, \ldots, N - 1\]

\[u_i \in U, \quad i = 0, \ldots, N - 1, \quad F(x_N) = 0\]
Example: Zermelo’s Problem

• Designing direct methods in Matlab: transcribe optimal control problem into a non-linear program, and solve it via \texttt{fmincon}

Modified Zermelo’s Problem

\begin{align*}
\text{(OCP)} & \quad \min \int_{0}^{t_f} u(t)^2 \, dt \\
& \quad \dot{x}(t) = v \cos(u(t)) + \text{flow}(y(t)), \ t \in [0, t_f] \\
& \quad \dot{y}(t) = v \sin(u(t)), \ t \in [0, t_f] \\
& \quad (x, y)(0) = 0, \ (x, y)(t_f) = (M, \ell) \\
& \quad |u(t)| \leq u_{\text{max}}, \ t \in [0, t_f]
\end{align*}

\begin{align*}
\text{(NLOP)} & \quad \min_{(x_i, u_i)} \sum_{i=0}^{N-1} h \ u_i^2 \\
& \quad x_{i+1} = x_i + h (v \cos(u_i) + \text{flow}(y_i)) \\
& \quad y_{i+1} = y_i + h v \sin(u_i), \ |u_i| \leq u_{\text{max}} \\
& \quad (x_0, y_0) = 0, \ (x_N, y_N) = (M, \ell)
\end{align*}

State and control parameterization method
**Results**

\[ |u(t)| \leq 1 \]  
(effectively, no control)  
\[ N = 20 \]  
28 iterations

\[ |u(t)| \leq 0.75 \]  
\[ N = 20 \]  
23 iterations
Transcription into nonlinear programming
(control parametrization method)

\[ \min \int_0^{t_f} g(x(t), u(t), t) \, dt \]

\[ \dot{x}(t) = f(x(t), u(t), t), \quad t \in [0, t_f] \]

(OCP)

\[ x(0) = x_0, \quad x(t_f) \in M_f \]

\[ u(t) \in U \subseteq \mathbb{R}^m, \quad t \in [0, t_f] \]

Time and control discretization

1. Select a discretization \( 0 = t_0 < t_1 < \cdots < t_N = t_f \) for the interval \([0, t_f]\) and, for every \( i = 0, \ldots, N - 1 \), define \( u_i \sim u(t), \quad t \in [t_i, t_{i+1}] \).

2. By denoting \( h_i = t_{i+1} - t_i \), \((OCP)\) is transcribed into the following nonlinear, constrained optimization problem:

\[ \min_{u_i} \sum_{i=0}^{N-1} h_i g(x(t_i), u_i, t_i) \]

\[(NLOP-C)\]

\[ u_i \in U, \quad i = 0, \ldots, N - 1, \quad F(x(t_N)) = 0 \]

where each \( x(t_i) \) is recursively computed via

\[ x(t_{i+1}) = x(t_i) + h_i f(x(t_i), u_i, t_i), \quad i = 0, \ldots, N - 1 \]
Example: Zermelo’s Problem

Modified Zermelo’s Problem

\[
\begin{align*}
\text{(OCP)} & \quad \min \int_0^{t_f} u(t)^2 \, dt \\
\dot{x}(t) &= v \cos(u(t)) + \text{flow}(y(t)), \quad t \in [0, t_f] \\
\dot{y}(t) &= v \sin(u(t)), \quad t \in [0, t_f] \\
(x, y)(0) &= 0, \quad (x, y)(t_f) = (M, \ell) \\
|u(t)| &\leq 1, \quad t \in [0, t_f]
\end{align*}
\]

Control parameterization method

\[
\begin{align*}
\text{(NLOP-C)} & \quad \min_{u_i} \sum_{i=0}^{N-1} h u_i^2 \\
(x, y)(t_N) &= (M, \ell), \quad |u_i| \leq u_{\max} \\
\text{where, recursively:} & \\
x(t_N) &= x_0 + h \sum_{i=0}^{N-1} (v \cos(u_i) + \text{flow}(y(t_i))) \\
y(t_i) &= y_0 + h \sum_{j=0}^{i} v \sin(u_j)
\end{align*}
\]
Results

\[ |u(t)| \leq 1 \]
(effectively, no control)
N = 30
50 iterations

\[ |u(t)| \leq 0.75 \]
N = 30
16 iterations
Next time

• Direct collocation and SCP