SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

Collocation Methods

CMPT 419/983
Mo Chen
SFU Computing Science
7/10/2019

Collocation

$$\underset{s,q}{\text{minimize}} \quad h(s_N, t_N) + \sum_{i=0}^{N-1} c(s_i, q_i, t_i)(t_{i+1} - t_i)$$

subject to $\forall i \in \{0,1,\ldots,N-1\}$,

~~$s_{i+1} = s_i + f(s_i, q_i)(t_{i+1} - t_i)$~~

$g(s_i, q_i) \geq 0$

- No numerical integration

- Directly approximates $x(t)$ and $u(t)$
  - **Piecewise: eg. Hermite-Simpson method**
  - Global: eg. Pseudospectral methods

- Impose dynamics constraints at discrete time points (“collocation points”)

Hermite-Simpson Collocation

- Discretize time:

$$t_0 < t_1 < \cdots < t_N \coloneqq t_f, \qquad h \coloneqq t_{i+1} - t_i$$

$$x_i \coloneqq x(t_i), \qquad u_i = u(t_i)$$

x_i and u_i are decision variables

- (Assume scalar x for now, and) write

$$x(t) = b_{i,0} + b_{i,1}(t - t_i) + b_{i,2}(t - t_i)^2 + b_{i,3}(t - t_i)^3, \qquad t \in [t_i, t_{i+1}]$$

$$\dot{x}(t) = b_{i,1} + 2b_{i,2}(t - t_i) + 3b_{i,3}(t - t_i)^2, \qquad t \in [t_i, t_{i+1}]$$

1. Some algebra: At t_i and t_{i+1}:

$$\begin{bmatrix} x(t_i) \\ \dot{x}(t_i) \\ x(t_{i+1}) \\ \dot{x}(t_{i+1}) \end{bmatrix} = \begin{bmatrix} x_i \\ f(x_i, u_i) \\ x_{i+1} \\ f(x_{i+1}, u_{i+1}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & h & h^2 & h^3 \\ 0 & 1 & 2h & 3h^2 \end{bmatrix} \begin{bmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \\ b_{i,3} \end{bmatrix}$$

- Obtain coefficients in terms of decision variables by taking inverse

$$\begin{bmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \\ b_{i,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/h^2 & -2/h & 3/h^2 & -1/h \\ 2/h^3 & 1/h^2 & -2/h^3 & 1/h^2 \end{bmatrix} \begin{bmatrix} x_i \\ f(x_i, u_i) \\ x_{i+1} \\ f(x_{i+1}, u_{i+1}) \end{bmatrix}$$

Dynamics Constraint

2. Choice of collocation points:

$$t_{i,c} = \frac{t_i + t_{i+1}}{2}$$

$$u_{i,c} := \frac{u_{i+1} + u_i}{2}$$

- Plug in $t_{i,c}$:

$$x_{i,c} = b_{i,0} + b_{i,1}(t_{i,c} - t_i) + b_{i,2}(t_{i,c} - t_i)^2 + b_{i,3}(t_{i,c} - t_i)^3$$
$$\dot{x}_{i,c} = b_{i,1} + 2b_{i,2}(t_{i,c} - t_i) + 3b_{i,3}(t_{i,c} - t_i)^2$$

3. Dynamics constraint at collocation points:

$$\dot{x}_{i,c} - f(x_{i,c}, u_{i,c}) = 0$$

- $x_{i,c}, \dot{x}_{i,c}$ depend on $b_{i,0}, b_{i,1}, b_{i,2}, b_{i,3}$
- $b_{i,0}, b_{i,1}, b_{i,2}, b_{i,3}$ depend on $x_i, x_{i+1}, u_i, u_{i+1}$
- $u_{i,c}$ depends on u_i, u_{i+1}

$$\begin{bmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \\ b_{i,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/h^2 & -2/h & 3/h^2 & -1/h \\ 2/h^3 & 1/h^2 & -2/h^3 & 1/h^2 \end{bmatrix} \begin{bmatrix} x_i \\ f(x_i, u_i) \\ x_{i+1} \\ f(x_{i+1}, u_{i+1}) \end{bmatrix}$$

Hermite-Simpson collocation

- Optimization problem, with simple integration

$$\underset{\{x_i\}_{i=1}^{N},\{u_i\}_{i=1}^{N-1}}{\text{minimize}} \quad h(x_N, t_N) + \sum_{i=0}^{N-1} c(x_i, u_i, t_i)(t_{i+1} - t_i)$$

subject to $\forall i \in \{0,1,\dots,N-1\}$,

$$\dot{x}_{i,c} - f(x_{i,c}, u_{i,c}) = 0$$

$$g(x_i, u_i) \geq 0$$

$$x_{i,c} = b_{i,0} + b_{i,1}(t_{i,c} - t_i) + b_{i,2}(t_{i,c} - t_i)^2 + b_{i,3}(t_{i,c} - t_i)^3$$

$$\dot{x}_{i,c} = b_{i,1} + 2b_{i,2}(t_{i,c} - t_i) + 3b_{i,3}(t_{i,c} - t_i)^2$$

$$\begin{bmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \\ b_{i,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/h^2 & -2/h & 3/h^2 & -1/h \\ 2/h^3 & 1/h^2 & -2/h^3 & 1/h^2 \end{bmatrix} \begin{bmatrix} x_i \\ f(x_i, u_i) \\ x_{i+1} \\ f(x_{i+1}, u_{i+1}) \end{bmatrix}$$

$$u_{i,c} = \frac{u_{i+1} + u_i}{2}$$

Hermite-Simpson collocation

- Optimization problem, with simple integration

$$\underset{\{x_i\}_{i=1}^{N},\{u_i\}_{i=1}^{N-1}}{\text{minimize}} \quad h(x_N, t_N) + \sum_{i=0}^{N-1} c(x_i, u_i, t_i)(t_{i+1} - t_i)$$

$$\text{subject to} \quad \forall i \in \{0,1,\dots,N-1\},$$

$$\boldsymbol{\dot{x}_{i,c} - f\left(x_{i,c}, u_{i,c}\right) = 0}$$

$$g(x_i, u_i) \geq 0$$

- Key difference from shooting methods
  - Dynamics constraint: no numerical integration

Pseudospectral Methods

- Represent entire state trajectory as sum of weighted basis functions
  - Chebyshev polynomials, Legendre polynomials, etc.

- Pros:
  - Fewer decision variables
  - Numerically more accurate

- Cons:
  - Dense optimization problems

Receding Horizon Control

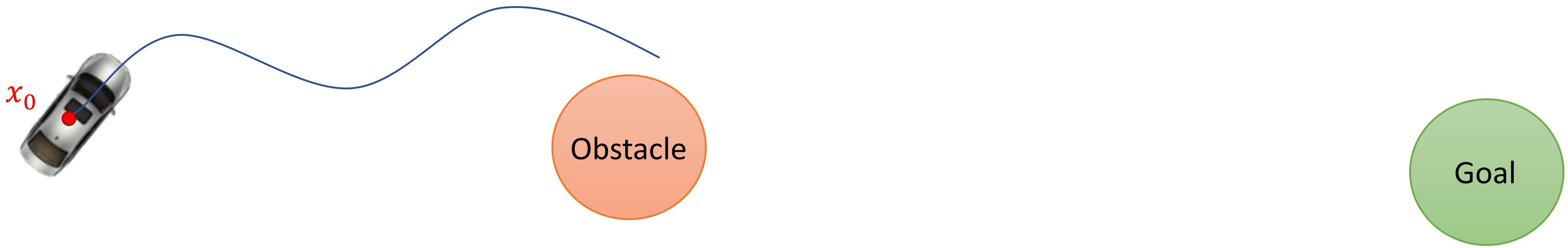
$$\underset{q}{\text{minimize}} \quad l(x(t_N), t_N) + \sum_{i=0}^{N-1} c(x(t_i), q_i, t_i)(t_{i+1} - t_i)$$

$$\text{subject to} \quad \forall i \in \{0,1,\dots,N-1\},$$

$$x(t_{i+1}) = x(t_i) + f(x(t_i), q_i)(t_{i+1} - t_i)$$

$$g(x(t_i), q_i) \geq 0$$

- Start from x_0, initial state; solve optimization
  - q provides control from time steps 0 to $N-1$ ← not necessary a long time horizon
  - Apply control only at time step 0



Receding Horizon Control

$$\underset{q}{\text{minimize}} \quad l(x(t_N), t_N) + \sum_{i=0}^{N-1} c(x(t_i), q_i, t_i)(t_{i+1} - t_i)$$

$$\text{subject to} \quad \forall i \in \{0,1,\dots,N-1\},$$
$$x(t_{i+1}) = x(t_i) + f(x(t_i), q_i)(t_{i+1} - t_i)$$
$$g(x(t_i), q_i) \geq 0$$

- Start from x_0, initial state; solve optimization
  - q provides control from time steps 0 to $N-1$ ← not necessary a long time horizon
  - Apply control only at time step 0
- Now, the state is at $x(t_{i+1})$; re-solve the optimization
  - Obtain control from time steps $i+1$ to $i+N$

Receding Horizon Control

$$\underset{q}{\text{minimize}} \quad l(x(t_N), t_N) + \sum_{i=0}^{N-1} c(x(t_i), q_i, t_i)(t_{i+1} - t_i)$$

$$\text{subject to} \quad \forall i \in \{0,1,\dots,N-1\},$$

$$x(t_{i+1}) = x(t_i) + f(x(t_i), q_i)(t_{i+1} - t_i)$$

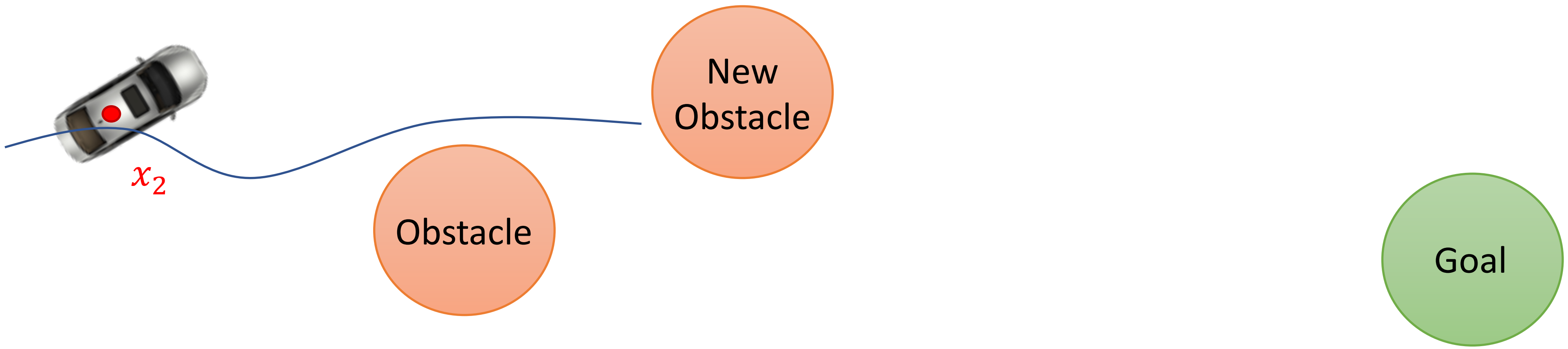
$$g(x(t_i), q_i) \geq 0$$

- Start from x_0, initial state; solve optimization
  - q provides control from time steps 0 to $N-1$ ← not necessary a long time horizon
  - Apply control only at time step 0
- Now, the state is at $x(t_{i+1})$; re-solve the optimization
  - Obtain control from time steps $i+1$ to $i+N$
  - Apply control at time step $i+1$

Receding Horizon Control

$$\underset{q}{\text{minimize}} \quad l(x(t_N), t_N) + \sum_{i=0}^{N-1} c(x(t_i), q_i, t_i)(t_{i+1} - t_i)$$
$$\text{subject to} \quad \forall i \in \{0,1,\dots,N-1\},$$
$$x(t_{i+1}) = x(t_i) + f(x(t_i), q_i)(t_{i+1} - t_i)$$
$$g(x(t_i), q_i) \geq 0$$

- Start from x_0, initial state; solve optimization
  - q provides control from time steps 0 to $N-1$ ← not necessary a long time horizon
  - Apply control only at time step 0
- Now, the state is at $x(t_{i+1})$; re-solve the optimization
  - Obtain control from time steps $i+1$ to $i+N$
  - Apply control at time step $i+1$
  - Repeat



Receding Horizon Control

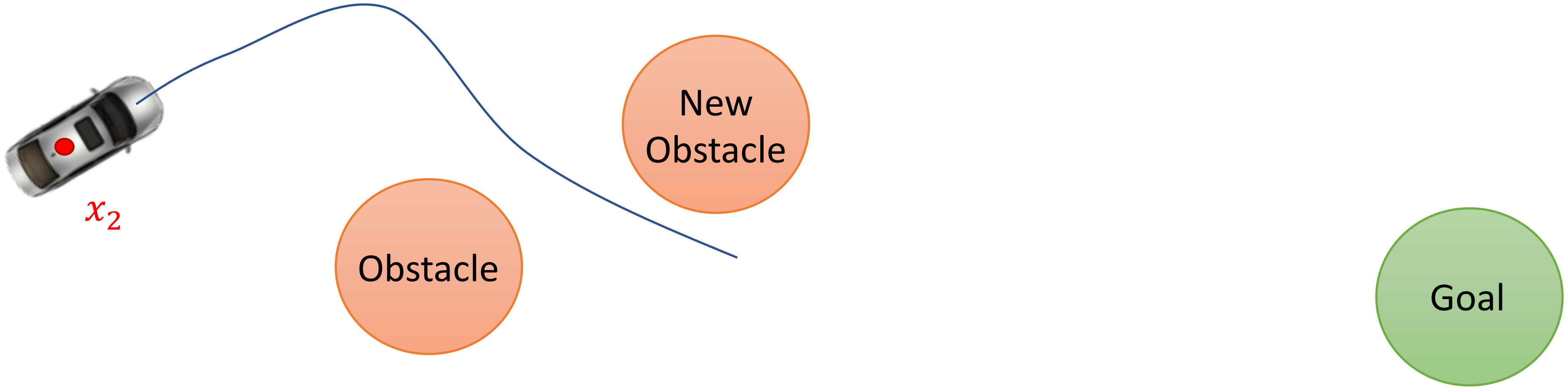
$$\underset{q}{\text{minimize}} \quad l(x(t_N), t_N) + \sum_{i=0}^{N-1} c(x(t_i), q_i, t_i)(t_{i+1} - t_i)$$

$$\text{subject to} \quad \forall i \in \{0,1,\dots,N-1\},$$

$$x(t_{i+1}) = x(t_i) + f(x(t_i), q_i)(t_{i+1} - t_i)$$

$$g(x(t_i), q_i) \geq 0$$

- Start from x_0, initial state; solve optimization
  - q provides control from time steps 0 to $N-1$ ← not necessary a long time horizon
  - Apply control only at time step 0
- Now, the state is at $x(t_{i+1})$; re-solve the optimization
  - Obtain control from time steps $i+1$ to $i+N$
  - Apply control at time step $i+1$
  - Repeat



Receding Horizon Control

- Main requirement
  - Computation must be fast enough compared to re-planning frequency
  - Re-planning frequency varies greatly depending on application
    - Agile mobile vehicles: milliseconds to a second
    - Building temperature control: minutes to hours

- Theoretical considerations
  - Recursive feasibility: feasible first optimization problem $\Rightarrow$ feasible kth optimization problem
  - Performance guarantees: eg. goal satisfaction

- Special popular case
  - Model-predictive control: uses a model of the system

Optimal Control

- Open-loop solutions
  - Differential flatness
  - Shooting methods
  - Collocation

- Receding horizon control:
  - Apply first part of the open-loop solution
  - Resolve open-loop optimization

- Relevant software packages
  - Optimization: cvx, Gurobi, SeDuMi, Mosek, Cplex, Matlab (`fmincon`)
  - Shooting/collocation: casadi, ACADO, Matlab `bvp4c` (and similar)
  - Receding horizon control: ACADO, Matlab (MPC toolbox)