Direct Collocation via Trapezoid Method

Matthew P. Kelly

March 24, 2018

This document gives a few examples of how to perform transcription using the trapezoid method for direct collocation.

1 Boundary Value Problem with Path Objective (simple)

Here is a general formulation for a continuous-time trajectory optimization problem. For now, let's assume that the duration T is given.

$$
\begin{aligned}
&\text{minimize:} && J = \int_0^T g(t,\, \boldsymbol{x},\, \boldsymbol{u})\, dt \\
&\text{subject to:} && \mathbf{0} = \boldsymbol{h}(\boldsymbol{x}(0),\, \boldsymbol{x}(T)) \\
&\text{dynamics:} && \dot{\boldsymbol{x}} = \boldsymbol{f}(t,\, \boldsymbol{x},\, \boldsymbol{u})
\end{aligned}
$$

Perform direct transcription using the trapezoid method. In other words, the system dynamics and path objective are both approximated using the trapezoid rule for quadrature. To start with, we will us a uniform time grid with N segments:

$$
h = \frac{T}{N} \qquad\qquad t_k = k \cdot h
$$

Decision Variables: $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \quad \boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$

Objective Function: $J = \dfrac{h}{2} \cdot \displaystyle\sum_{k=0}^{N-1} \Big(g(t_k,\, \boldsymbol{x}_k,\, \boldsymbol{u}_k) \,+\, g(t_{k+1},\, \boldsymbol{x}_{k+1},\, \boldsymbol{u}_{k+1}) \Big)$

Boundary Constraints: $\mathbf{0} = \boldsymbol{h}(\boldsymbol{x}_0,\, \boldsymbol{x}_N)$

System Dynamics Constraints:

$$
\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{h}{2} \cdot \Big(\boldsymbol{f}(t_k,\, \boldsymbol{x}_k,\, \boldsymbol{u}_k) \,+\, \boldsymbol{f}(t_{k+1},\, \boldsymbol{x}_{k+1},\, \boldsymbol{u}_{k+1}) \Big) \qquad\qquad k \in 0 \ldots (N-1)
$$

2 Boundary Value Problem with Path Objective (Arbitrary Grid)

Here is a general formulation for a continuous-time trajectory optimization problem. Now let's make the boundary times decision variables and use an arbitrary grid.

$$\text{minimize:} \qquad J = \int_{T_0}^{T_F} g(t,\, \boldsymbol{x},\, \boldsymbol{u})\, dt$$

$$\text{subject to:} \qquad \boldsymbol{0} = \boldsymbol{h}(T_0,\, T_F,\, \boldsymbol{x}(T_0),\, \boldsymbol{x}(T_F))$$

$$\text{dynamics:} \qquad \dot{\boldsymbol{x}} = \boldsymbol{f}(t,\, \boldsymbol{x},\, \boldsymbol{u})$$

We will use an arbitrary grid, which we assume is provided either by the user or a mesh refinement algorithm. We then apply a linear mapping to this initial grid such that the initial and final points on the grid match up with the initial and final times that the optimization is using. This allows us to compute the time grid from the boundary times (t_0, t_N) and the mesh fraction $(\gamma_0, \gamma_1, \ldots, \gamma_N)$.

$$h_k = t_{k+1} - t_k \qquad\qquad t_k = \frac{t_N - t_0}{\gamma_N - \gamma_0} \cdot (\gamma_k - \gamma_0) + t_0$$

Decision Variables: $t_0,\, t_N, \quad \boldsymbol{x}_0,\, \boldsymbol{x}_1,\, \ldots,\, \boldsymbol{x}_N, \quad \boldsymbol{u}_0,\, \boldsymbol{u}_1,\, \ldots,\, \boldsymbol{u}_N$

Objective Function: $J = \frac{1}{2} \cdot \sum_{k=0}^{N-1} h_k \Big(g(t_k,\, \boldsymbol{x}_k,\, \boldsymbol{u}_k) \,+\, g(t_{k+1},\, \boldsymbol{x}_{k+1},\, \boldsymbol{u}_{k+1}) \Big)$

Boundary Constraints: $\boldsymbol{0} = \boldsymbol{h}(t_0,\, t_N,\, \boldsymbol{x}_0,\, \boldsymbol{x}_N)$

System Dynamics Constraints:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{h_k}{2} \cdot \Big(\boldsymbol{f}(t_k,\, \boldsymbol{x}_k,\, \boldsymbol{u}_k) \,+\, \boldsymbol{f}(t_{k+1},\, \boldsymbol{x}_{k+1},\, \boldsymbol{u}_{k+1}) \Big) \qquad\qquad k \in 0 \ldots (N-1)$$