About me

Haichao Hong / 洪海超

Associate Professor

Email: haichao.hong@sjtu.edu.cn

Office: A227

Homepage: haichao.de

Contents

A guidance problem:

$$
\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}\left(\boldsymbol{x}\left(t\right), \boldsymbol{u}\left(t\right), \boldsymbol{p}, t\right) \ , \\
\boldsymbol{x}(t_0) &= \boldsymbol{x}_1 \ , \\
\boldsymbol{x}(t_f) &= \boldsymbol{x}_f
\end{aligned}
$$

- Initial condition fixed
- Terminal condition fixed
- No path constraints
- To determine a control sequence $\boldsymbol{u}(t)$

With a known $\boldsymbol{x}_1$, using N discrete steps, step length h

To determine a control sequence

$$\boldsymbol{U} = [\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, ..., \boldsymbol{u}_{N-1}^T]^T \in \mathbb{R}^{(N-1)m}$$

Using Euler Forward: (Number of states is n)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + h\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ &= \boldsymbol{F}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \end{aligned}$$

With a known $\boldsymbol{x}_1$, using N discrete steps, step length h

To determine a control sequence

$$\boldsymbol{U}=[\boldsymbol{u}_1^T,\boldsymbol{u}_2^T,\ldots,\boldsymbol{u}_{N-1}^T]^T\in\mathbb{R}^{(N-1)m}$$

Using Euler Forward: (Number of states is n)

$$\begin{aligned}
\boldsymbol{x}_2 &= \boldsymbol{F}_1\left(\boldsymbol{x}_1,\boldsymbol{u}_1\right)\\
\boldsymbol{x}_3 &= \boldsymbol{F}_2\left(\boldsymbol{F}_1\left(\boldsymbol{x}_1,\boldsymbol{u}_1\right),\boldsymbol{u}_2\right)\\
&\vdots\\
\boldsymbol{x}_N &= \boldsymbol{F}_{N-1}\left(\boldsymbol{F}_{N-2}\left(\ldots\left(\boldsymbol{F}_1\left(\boldsymbol{x}_1,\boldsymbol{u}_1\right),\ldots\right),\boldsymbol{u}_{N-2}\right),\boldsymbol{u}_{N-1}\right)
\end{aligned}$$

A guidance problem:

$$
\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}\left(\boldsymbol{x}\left(t\right), \boldsymbol{u}\left(t\right), \boldsymbol{p}, t\right)\ , \\
\boldsymbol{x}(t_0) &= \boldsymbol{x}_1\ , \\
\boldsymbol{x}(t_f) &= \boldsymbol{x}_f
\end{aligned}
\quad \Longrightarrow \quad \boldsymbol{x}_N = \boldsymbol{x}_f
$$

$$\Downarrow$$

$$\boldsymbol{x}_N = \boldsymbol{F}_{N-1}\left(\boldsymbol{F}_{N-2}\left(\ldots\left(\boldsymbol{F}_1\left(\boldsymbol{x}_1, \boldsymbol{u}_1\right), \ldots\right), \boldsymbol{u}_{N-2}\right), \boldsymbol{u}_{N-1}\right) = \boldsymbol{x}_f$$

$$\Downarrow$$

$$\boldsymbol{F}_{N-1}\left(\boldsymbol{F}_{N-2}\left(\ldots\left(\boldsymbol{F}_1\left(\boldsymbol{x}_1, \boldsymbol{u}_1\right), \ldots\right), \boldsymbol{u}_{N-2}\right), \boldsymbol{u}_{N-1}\right) - \boldsymbol{x}_f = 0$$

$$\boldsymbol{F}_{N-1}\left(\boldsymbol{F}_{N-2}\left(\ldots\left(\boldsymbol{F}_1\left(\boldsymbol{x}_1, \boldsymbol{u}_1\right), \ldots\right), \boldsymbol{u}_{N-2}\right), \boldsymbol{u}_{N-1}\right) - \boldsymbol{x}_f = 0$$

Unknown: $\boldsymbol{U} = [\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_{N-1}^T]^T \in \mathbb{R}^{(N-1)m}$

Number of equations: n (Number of states)

It is very likely that $(N-1)m \gg n$

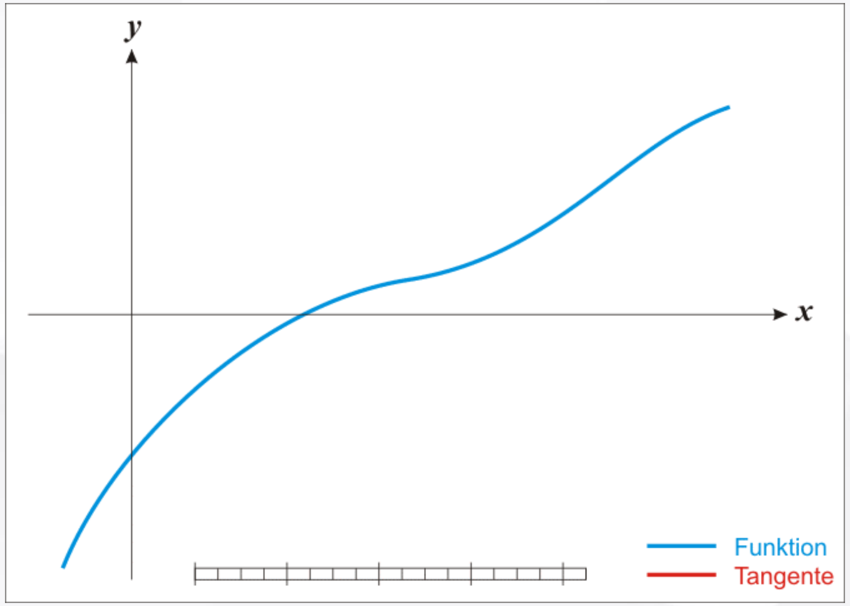
$$\boldsymbol{G}(\boldsymbol{U}) = 0$$

It is an underdetermined system! But Nonlinear

To find the root of a nonlinear underdetermined system

$$\boldsymbol{G}\left(\boldsymbol{U}\right) = 0$$

Newton's method is available for this task:



To find the root of a nonlinear underdetermined system

$$\boldsymbol{G}(\boldsymbol{U}) = 0$$

Newton's method is available for this task:

$$\boldsymbol{U} = \boldsymbol{U}^p + d\boldsymbol{U} \qquad \boldsymbol{r} - \boldsymbol{G}'(\boldsymbol{U}^p)d\boldsymbol{U} = 0$$

where $\boldsymbol{G}'$ is the Jacobian matrix, $d\boldsymbol{U}$ is the Newton step, and $\boldsymbol{r}$ is the residual from the previous iteration.

$$\boldsymbol{r} - \boldsymbol{G}^{'}(\boldsymbol{U}^{p})d\boldsymbol{U} = 0$$

$\boldsymbol{U}^{p}$ is the previous solution of the iterations.

This means that we need **an initial solution**.

It is an underdetermined system again! But Linear

We can find the least-squares solution directly.

- MATLAB function mldivide A\B

The Jacobian matrix is computed as

$$\boldsymbol{G}^{'}(\boldsymbol{U}) = \begin{bmatrix} \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{u}_1} & \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{u}_2} & \cdots & \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{u}_{N-1}} \end{bmatrix}$$

where

$$\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{u}_i} = \left[\frac{\partial \boldsymbol{F}_{N-1}}{\partial \boldsymbol{x}_{N-1}}\right]\left[\frac{\partial \boldsymbol{F}_{N-2}}{\partial \boldsymbol{x}_{N-2}}\right]\cdot \ldots \cdot \left[\frac{\partial \boldsymbol{F}_{i+1}}{\partial \boldsymbol{x}_{i+1}}\right]\left[\frac{\partial \boldsymbol{F}_{i}}{\partial \boldsymbol{u}_{i}}\right],$$

$$i = 1, \ldots, N-1,$$

For using the MATLAB function mldivide A\B,

by default, the least-squares method minimizes

$$J = \frac{1}{2} d\boldsymbol{U}^T d\boldsymbol{U}$$

- Unclear physical meaning

A different cost leads to clearer physical meaning

$$J = \frac{1}{2} \boldsymbol{U}^T \boldsymbol{R} \boldsymbol{U}$$

Minimize the control effort

$$= \frac{1}{2} \left(d\boldsymbol{U} + \boldsymbol{U}^p \right)^T \boldsymbol{R} \left(d\boldsymbol{U} + \boldsymbol{U}^p \right)$$

Newton-Type Methods in Computational Guidance

$$\underset{d\boldsymbol{U}}{\text{minimize}} \quad J = \frac{1}{2}\left(d\boldsymbol{U} + \boldsymbol{U}^p\right)^T \boldsymbol{R}\left(d\boldsymbol{U} + \boldsymbol{U}^p\right)$$

$$\text{subject to} \quad \boldsymbol{r} - \boldsymbol{G}^{'}\left(\boldsymbol{U}^p\right) d\boldsymbol{U} = 0$$

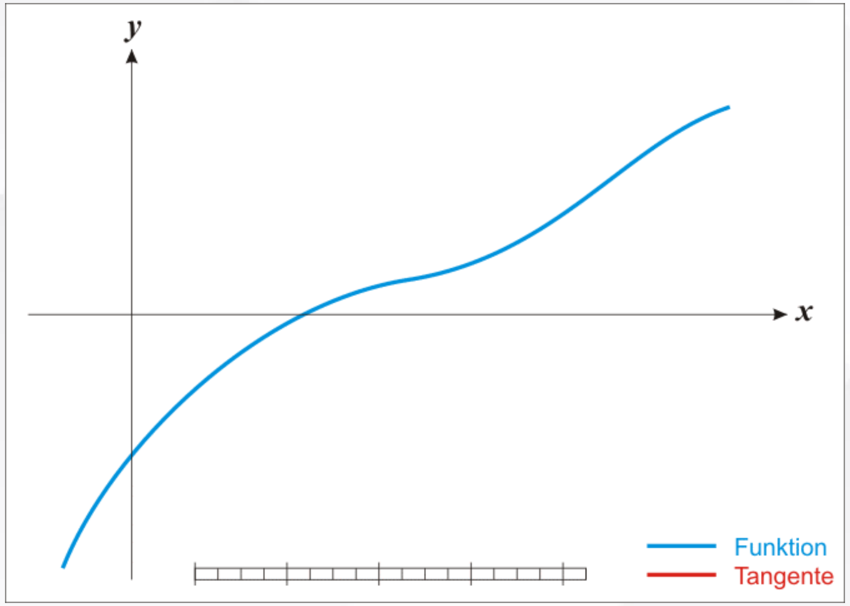
Introducing a Lagrange multiplier

$$\mathcal{L} = \frac{1}{2}(d\boldsymbol{U} + \boldsymbol{U}^p)^T \boldsymbol{R}(d\boldsymbol{U} + \boldsymbol{U}^p) + \boldsymbol{\lambda}^T \left[\boldsymbol{r} - \boldsymbol{G}^{'}(\boldsymbol{U}^p) d\boldsymbol{U}\right],$$

Using the optimality conditions

$$\frac{\partial \mathcal{L}}{\partial (d\boldsymbol{U})} = 0 \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = 0$$

One more step!



Newton iterations $\boldsymbol{U} = \boldsymbol{U}^p + d\boldsymbol{U} \rightarrow \boldsymbol{U}^p$

Use the new $\boldsymbol{U}$ to get new $\boldsymbol{x}_k, \;\; k = 1, 2, \ldots, N$

$$\frac{\partial G}{\partial \boldsymbol{u}_i} = \left[\frac{\partial \boldsymbol{F}_{N-1}}{\partial \boldsymbol{x}_{N-1}}\right]\left[\frac{\partial \boldsymbol{F}_{N-2}}{\partial \boldsymbol{x}_{N-2}}\right] \cdots \cdot \left[\frac{\partial \boldsymbol{F}_{i+1}}{\partial \boldsymbol{x}_{i+1}}\right]\left[\frac{\partial \boldsymbol{F}_{i}}{\partial \boldsymbol{u}_{i}}\right],$$

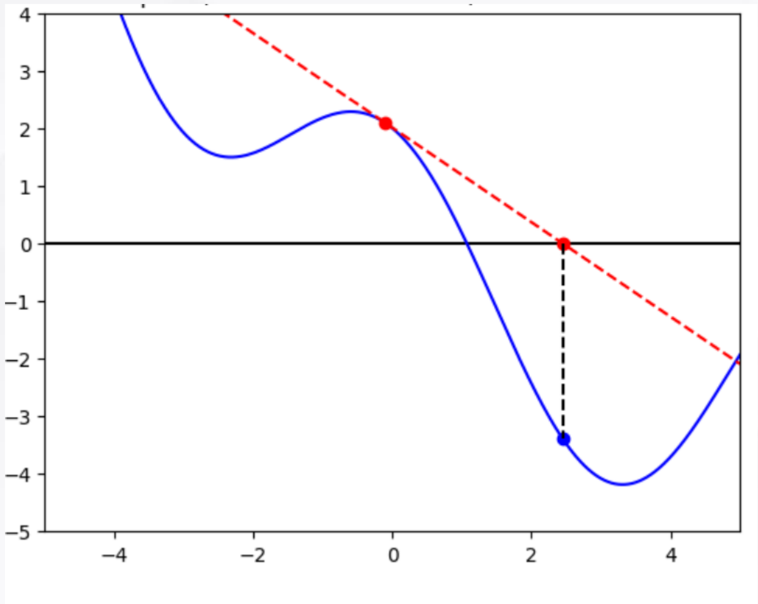
$$i = 1, \ldots, N-1,$$

One more step!

Line Search

Newton iterations $\boldsymbol{U} = \boldsymbol{U}^p + s \cdot d\boldsymbol{U}$

Improve convergence



Algorithm 1 Line-Search Strategy

Get $\boldsymbol{U}^p$, $d\boldsymbol{U}$, $s = 1$, $\kappa \in (0, 1]$
while $\left|\boldsymbol{G}_n\left(\boldsymbol{U}^p + s \cdot d\boldsymbol{U}\right)\right|_\infty > \left|\boldsymbol{G}_n\left(\boldsymbol{U}^p\right)\right|_\infty$ **do**
  $s \leftarrow \kappa s$
end while
$\boldsymbol{U} \leftarrow \boldsymbol{U}^p + s \cdot d\boldsymbol{U}$
return $\boldsymbol{U}$

Found the root of a nonlinear underdetermined system

$$\boldsymbol{G}\left(\boldsymbol{U}\right) = 0$$

Solve the guidance problem with a minimum control effort

$$\begin{aligned}
&\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}\left(t\right), \boldsymbol{u}\left(t\right), \boldsymbol{p}, t\right) \ , \\
&\boldsymbol{x}(t_0) = \boldsymbol{x}_1 \ , \\
&\boldsymbol{x}(t_f) = \boldsymbol{x}_f
\end{aligned}$$

No sacrifice of nonlinearity

Re-visit with a variable step length

With a known $\boldsymbol{x}_1$, using N discrete steps, step length h

To determine a control sequence

$$\boldsymbol{U} = [\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, ..., \boldsymbol{u}_{N-1}^T, h]^T \in \mathbb{R}^{(N-1)m+1}$$

Using Euler Forward:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + h\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ &= \boldsymbol{F}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, h) \end{aligned}$$

With a known $\boldsymbol{x}_1$, using N discrete steps, step length h

To determine a control sequence

$$\boldsymbol{U} = [\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, ..., \boldsymbol{u}_{N-1}^T, h]^T \in \mathbb{R}^{(N-1)m+1}$$

Using Euler Forward:

$$\begin{aligned}
\boldsymbol{x}_2 &= \boldsymbol{F}_1(\boldsymbol{x}_1, \boldsymbol{u}_1, h) \\
\boldsymbol{x}_3 &= \boldsymbol{F}_2(\boldsymbol{F}_1(\boldsymbol{x}_1, \boldsymbol{u}_1, h), \boldsymbol{u}_2, h) \\
&\vdots \\
\boldsymbol{x}_N &= \boldsymbol{F}_{N-1}(\boldsymbol{F}_{N-2}(...(\boldsymbol{F}_1(\boldsymbol{x}_1, \boldsymbol{u}_1, h), ...), \boldsymbol{u}_{N-2}, h), \boldsymbol{u}_{N-1}, h)
\end{aligned}$$

$$\boldsymbol{F}_{N-1}\left(\boldsymbol{F}_{N-2}\left(\ldots\left(\boldsymbol{F}_{1}\left(\boldsymbol{x}_1, \boldsymbol{u}_1, h\right), \ldots\right), \boldsymbol{u}_{N-2}, h\right), \boldsymbol{u}_{N-1}, h\right) - \boldsymbol{x}_f = 0$$

Unknown: $\boldsymbol{U} = [\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_{N-1}^T, h]^T \in \mathbb{R}^{(N-1)m+1}$

Number of equations: n (Number of states)

It is very likely that $(N-1)m+1 \gg n$

$$\boldsymbol{G}\left(\boldsymbol{U}\right) = 0$$

It is an underdetermined system! But Nonlinear

Newton-Type Methods in Computational Guidance*:

- Un(path-)constrained trajectory optimization
- Newton method for finding the root of a nonlinear underdetermined system
- Lagrange multiplier for linearly constrained optimization
- Line search

Zheng. H., Piprek, P., Hong, H., Holzapfel, F., and Tang, S., “Smooth Sub-optimal Trajectory Generation for Transition Maneuvers,” IEEE Access, Vol. 8, pp. 61035–61042.
Pan B., Ma Y. and Yan R., “Newton-Type Methods in Computational Guidance,” Journal of Guidance, Control, and Dynamics, Vol. 42, No. 2, 2019, pp. 377–383.

Newton-Type Methods in Computational Guidance*:

- Was originally developed by Padhi and referred to as the Model Predictive Static Programming
- For unconstrained trajectory optimization problem
- Minimize the control effort
- Very computationally efficient
- Limited choice of cost function

Zheng. H., Piprek, P., Hong, H., Holzapfel, F., and Tang, S., “Smooth Sub-optimal Trajectory Generation for Transition Maneuvers,” IEEE Access, Vol. 8, pp. 61035–61042.
Pan B., Ma Y. and Yan R., “Newton-Type Methods in Computational Guidance,” Journal of Guidance, Control, and Dynamics, Vol. 42, No. 2, 2019, pp. 377–383.

Take-home messages behind Newton-Type Methods:

- We can develop dedicated tool for a specific problem.
- Tool can be just simple enough to solve the problem in order to be efficient.

What if constrained?

Homework

Tell me your understanding of “What is Guidance”

- Send to haichao.hong@sjtu.edu.cn
- Include “作业” and YOUR NAME in the subject line
- NO COPY PASTE