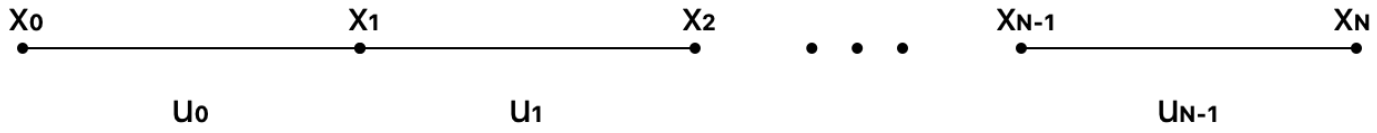
Optimal Control with GEKKO

Optimal Control

Suppose we have

- x - state variable $\in \mathbb{R}^{n_x}$
- u - control variable $\in \mathbb{R}^{n_u}$
- k - discrete time variable $\in \{0, \dots, N\}$, where $N \in \mathbb{N}$ is a time horizon



Define

- $\mathbf{x} = (x(0), x(1), \dots, x(N))$
- $\mathbf{u} = (u(0), u(1), \dots, u(N-1))$

Then the **Optimal Control Problem** is formulated as follows:

$$F(\mathbf{x}, \mathbf{u}) \to \min_{\mathbf{x},\mathbf{u}} \quad \begin{cases} x(0) = x_0 \\ x(k+1) = f(x(k), u(k)) \\ h(x(k), u(k)) \leq 0, k < N \\ \widehat{h}(x(N)) \leq 0 \end{cases} \tag{1}$$

where

- $F(\cdot) = V_f(x(N)) + \sum\limits_{k=0}^{N-1} l(x(k), u(k))$ - cost function
- $x(k+1) = f(x(k), u(k))$ - state equation

If f is linear, then the problem is called **Linear Optimal Control**. Otherwise it's called **Non-Linear Optimal Control**.

- $h(x(k), u(k)) \leq 0$ - path constraints
- $\widehat{h}(x(N)) \leq 0$ - final constraint

Mixed-Integer Optimal Control

Suppose in addition to a real control variable u we have an integer control variable i:

- $x \in \mathbb{R}^{n_x}$ - state variable
- $u \in \mathbb{R}^{n_u}$ - continuous control variable
- $i \in \mathbb{Z}^{n_i}$ - integer control variable inside a bounded convex polyhedron P

Define

- $\mathbf{x} = (x(0), x(1), \ldots, x(N))$
- $\mathbf{u} = (u(0), u(1), \ldots, u(N-1))$
- $\mathbf{i} = (i(0), i(1), \ldots, i(N-1))$

Then the **Mixed-Integer Optimal Control** problem is formulated as follows:

$$F(\mathbf{x}, \mathbf{u}, \mathbf{i}) \to \min_{\mathbf{x},\mathbf{u},\mathbf{i}} \quad \begin{cases} x(0) = x_0 \\ x(k+1) = f(x(k), u(k), i(k)) \\ h(x(k), u(k), i(k)) \leq 0, k < N \\ \widehat{h}(x(N)) \leq 0 \\ \mathbf{i} \in P \\ \mathbf{i} \in \mathbb{Z}^{N \cdot n_i} \end{cases} \tag{2}$$

Hereinafter we will assume that the objective terms consist of a nonlinear least squares term $\frac{1}{2}\|F_1(\cdot)\|_2^2$ and a nonlinear term $F_2(\cdot)$ - both are differentiable. Thus,

$$\begin{cases} F(\mathbf{x}, \mathbf{u}, \mathbf{i}) = V_f(x(N)) + \sum\limits_{k=0}^{N-1} l(x(k), u(k), i(k)) \\ \left.\begin{array}{l} l(x,u,i) = \frac{1}{2}\|l_1(x,u,i)\|_2^2 + l_2(x,u,i) \\ V_f(x) = \frac{1}{2}\|V_1(x)\|_2^2 + V_2(x) \end{array}\right\} \Rightarrow F(\mathbf{x}, \mathbf{u}, \mathbf{i}) = \frac{1}{2}\|F_1(\mathbf{x}, \mathbf{u}, \mathbf{i})\|_2^2 + F_2(\mathbf{x}, \mathbf{u}, \mathbf{i}) \end{cases} \tag{3}$$

Gauss-Newton algorithm

The proposal algorithm consists of three steps:

S1: solve the system without integrality constraint $\mathbf{i} \in \mathbb{Z}^{n_i}$:

$$(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) = \arg\min_{\mathbf{x},\,\mathbf{u},\,\mathbf{i}} F(\mathbf{x}, \mathbf{u}, \mathbf{i}), \quad \begin{cases} x(0) = x_0 \\ x(k+1) = f(x(k), u(k), i(k)) \\ h(x(k), u(k), i(k)) \leq 0, k < N \\ \widehat{h}(x(N)) \leq 0 \\ \mathbf{i} \in P \end{cases} \tag{4}$$

S2: approximate our continuous solution $\mathbf{i}^*$ by an integer $\mathbf{i}^{**}$ obtained as follows:

$$\mathbf{i}^{**} = \arg\min_{\mathbf{i}} d(\mathbf{i}, \mathbf{i}^*), \quad \begin{cases} x(0) = x_0 \\ x(k+1) = f_L(x(k), u(k), i(k)) \\ h_L(x(k), u(k), i(k)) \leqslant 0 \\ \widehat{h}_L(x(k)) \leqslant 0 \\ \mathbf{i} \in P \cap \mathbb{Z}^{N \cdot n_i} \end{cases} \tag{5}$$

Proposal Gauss-Newton algorithm - key part:

- $d(\mathbf{i}, \mathbf{i}^*) = J_{GN}(\mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) - J_{NLP}(\mathbf{i}^*)$ - distance function
- $J_{NLP}(\mathbf{i}^*) = F(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*)$

Gauss-Newton approximation part:

- $J_{GN}(\mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) = \min\limits_{\mathbf{x},\mathbf{u}} F_{GN}(\mathbf{x}, \mathbf{u}, \mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*)$
- $F_{GN}(\mathbf{x}, \mathbf{u}, \mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) = F_{QP}(\mathbf{x}, \mathbf{u}, \mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*, B_{GN}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*))$
- $F_{QP}(\mathbf{x}, \mathbf{u}, \mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*, B) = F_L(\mathbf{x}, \mathbf{u}, \mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) + \frac{1}{2} \begin{bmatrix} \mathbf{x} - \mathbf{x}^* \\ \mathbf{u} - \mathbf{u}^* \\ \mathbf{i} - \mathbf{i}^* \end{bmatrix}^T B \begin{bmatrix} \mathbf{x} - \mathbf{x}^* \\ \mathbf{u} - \mathbf{u}^* \\ \mathbf{i} - \mathbf{i}^* \end{bmatrix}$
- $B_{GN}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) = \frac{\partial F_1}{\partial(\mathbf{x},\mathbf{u},\mathbf{i})}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) \frac{\partial F_1}{\partial(\mathbf{x},\mathbf{u},\mathbf{i})}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*)^T$

Linearizations:

- $F_L(\mathbf{x}, \mathbf{u}, \mathbf{i} \mid \mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) = F(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*) + \frac{\partial F}{\partial(\mathbf{x},\mathbf{u},\mathbf{i})}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*)((\mathbf{x}, \mathbf{u}, \mathbf{i}) - (\mathbf{x}^*, \mathbf{u}^*, \mathbf{i}^*))$
- $f_L(x, u, i \mid x^*, u^*, i^*) = f(x^*, u^*, i^*) + \frac{\partial f}{\partial(x,u,i)}(x^*, u^*, i^*)((x, u, i) - (x^*, u^*, i^*))$
- $h_L(x, u, i \mid x^*, u^*, i^*) = h(x^*, u^*, i^*) + \frac{\partial h}{\partial(x,u,i)}(x^*, u^*, i^*)((x, u, i) - (x^*, u^*, i^*))$
- $\widehat{h}_L(x \mid x^*) = \widehat{h}(x^*) + \frac{\partial \widehat{h}}{\partial x}(x^*)(x - x^*)$

S3: solve the NLP system with fixed variable $\mathbf{i} = \mathbf{i}^{**}$:

$$(\mathbf{x}^{***}, \mathbf{u}^{***}, \mathbf{i}^{**}) = \arg\min_{\mathbf{x},\,\mathbf{u}} F(\mathbf{x}, \mathbf{u}, \mathbf{i}^{**}), \quad \begin{cases} x(0) = x_0 \\ x(k+1) = f(x(k), u(k), i^{**}(k)) \\ h(x(k), u(k), i^{**}(k)) \le 0, k < N \\ \widehat{h}(x(N)) \le 0 \end{cases} \tag{7}$$

Example

We will consider the problem from the article:

$$\begin{aligned} \text{continuous time}: &\quad \dot{x}(t) = f_c(x(t), u(t), i(t)) = x^3(t) - i(t) \\ \text{discrete time}: &\quad x(k+1) = f(x(k), u(k), i(k)) = \text{Runge-Kutta-4}\ (f_c) \end{aligned}$$

$$\text{objetive function}: F(\mathbf{x}, \mathbf{u}, \mathbf{i}) = \frac{1}{2} \sum_{k=0}^{N} (x(k) - x_{ref})^2$$

$$\mathbf{MINLP}: \min_{\mathbf{x},\mathbf{u},\mathbf{i}} F(\mathbf{x}, \mathbf{u}, \mathbf{i}), \quad \text{such that} \quad \begin{cases} x(0) = x_0 \\ x(k+1) = f(x(k), u(k), i(k)) \\ \mathbf{i} \in P \cap \mathbb{Z}^N \end{cases}$$

$$P = \left\{ \mathbf{i} \in [0,1]^N \;\middle|\; \begin{matrix} i(k) \ge i(k-1) - i(k-2) \\ i(k) \ge i(k-1) - i(k-3) \end{matrix} \right\}$$

Additional parameters: $N = 30,\ x_0 = 0.8,\ x_{ref} = 0.7$

Some remarks:

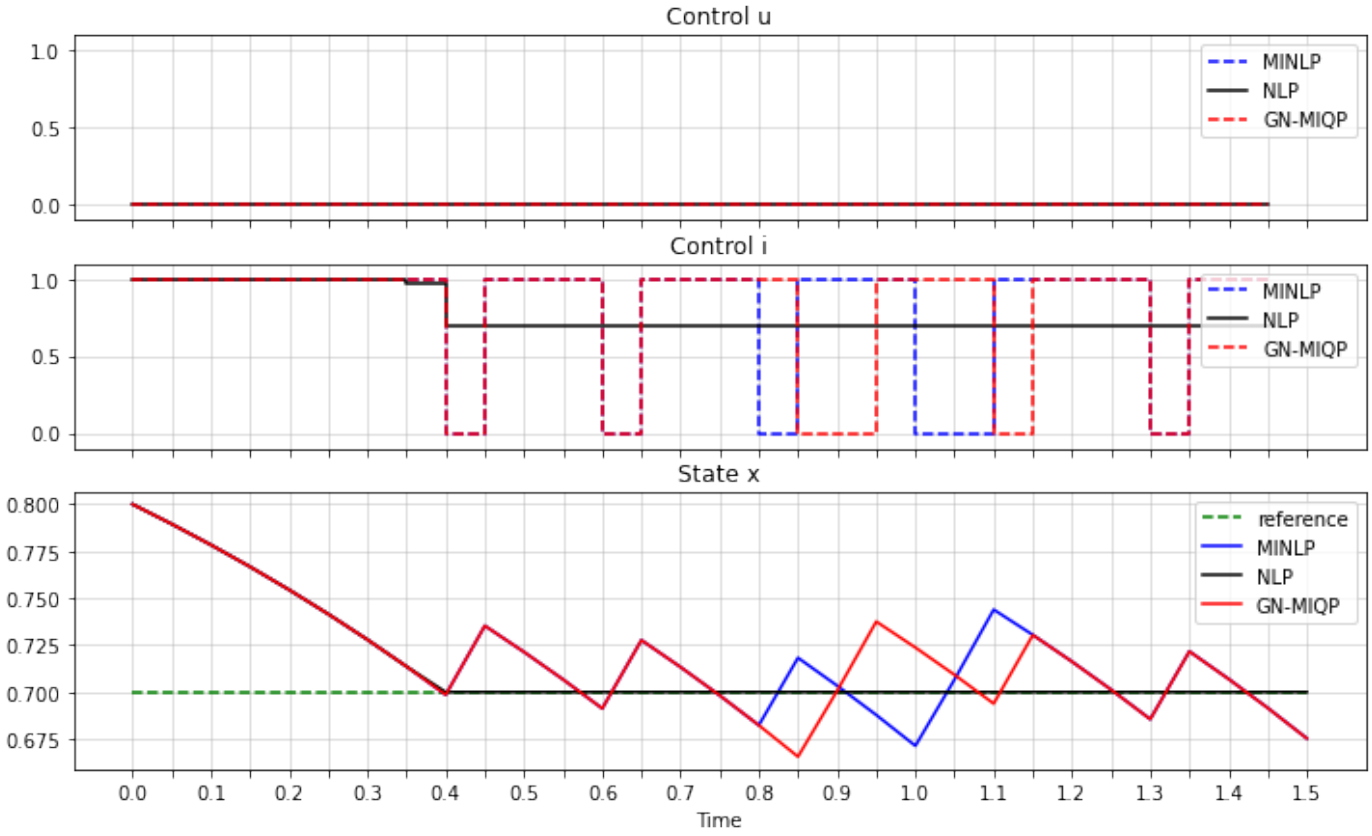
1. We can consider $F(\mathbf{x}, \mathbf{u}, \mathbf{i})$ as a sum of $F(x(k), u(k), i(k))$, where $F(x, u, i) = \frac{1}{2} \|F_1(x, u, i)\|_2^2$ and $F_1(x, u, i) = (x - x_{ref}, 0, 0)$.

2. Instead of Runge-Kutta methods we will use the collocations method. It's already implemented in GEKKO.
3. Firstly we will try $f(x, i) = x - i$ instead of $f(x, i) = x^3 - i$. Because in this case the Gauss-Newton system on the second step should be equal to the original system (linearization of f is equal to f if f is a linear function initially and quadratic approximation of F is equal to F if F is a quadratic function initially).

Results

Experiment 1

1. F is square
2. f is linear ($x - i$)

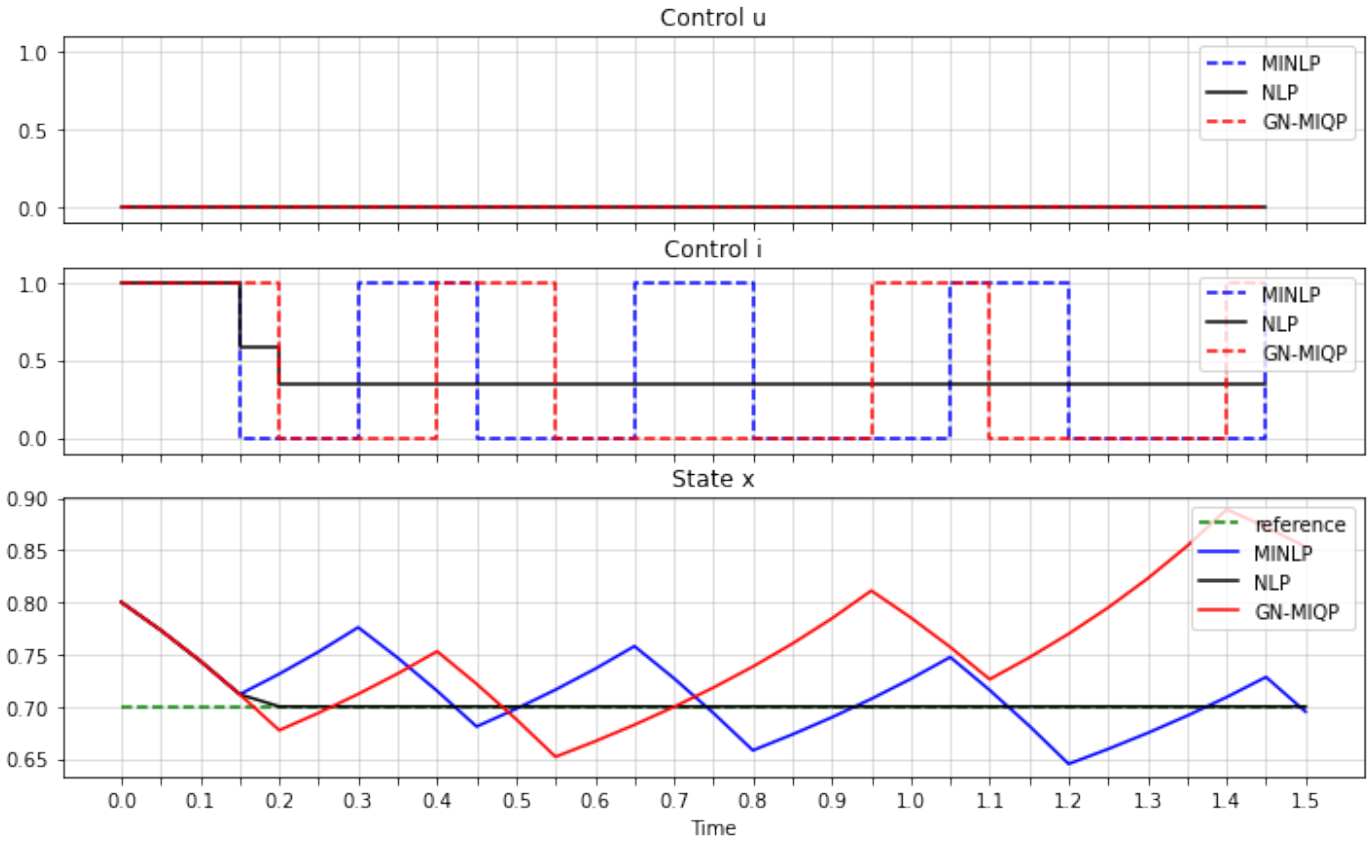



```
Objective value
---------------------
GEKKO MINLP : 0.01661
        NLP : 0.01213
    GN-MIQP : 0.01660
---------------------
GN-MIQP is equal to GEKKO MINLP
```

In this case the results should be equal because F_{GN} coincides with F and f_L coincides with f.

Experiment 2

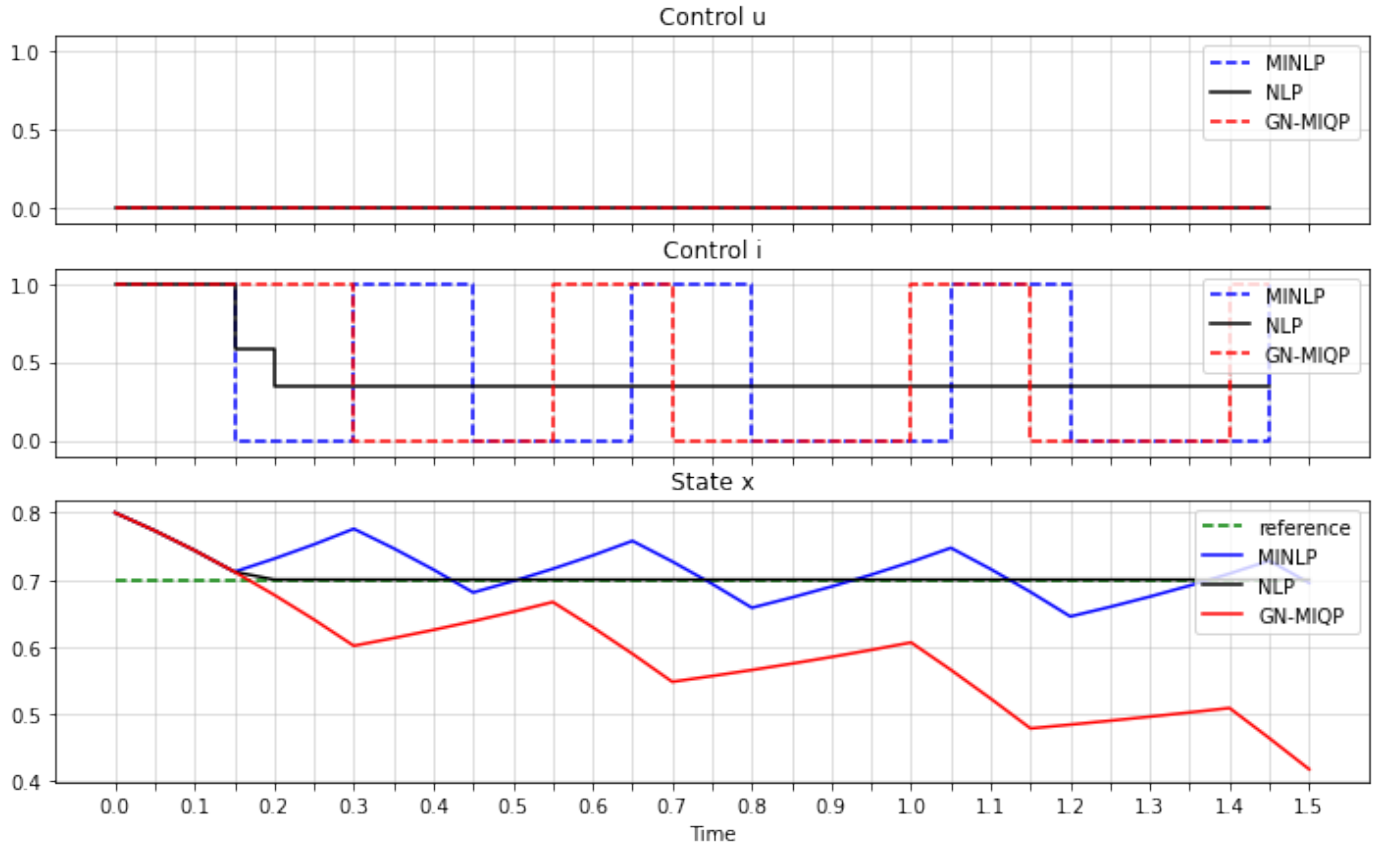
1. F is square
2. f is non-linear ($x^3 - i$)



We can see that the red line (on the bottom state graph) corresponded to Gauss-Newton method increases, moving away from the reference value. Since the objective function F is square, F_{GN} coincides with it. So, the only reason the solutions differ is because of the function f_L doesn't coincide with f. Not surprisingly, because approximating a nonlinear function with a linear one leads to errors.

```
Objective value
---------------------
GEKKO MINLP : 0.01892
        NLP : 0.00369
    GN-MIQP : 0.09791
---------------------
GN-MIQP is worse than GEKKO MINLP
```

Moreover, for this case the authors propose an exact solution. Let's substitute this solution.



Unfortunately, it gives even worse results than the GEKKO solution.