A Minimum Working Example for the Gauss-Newton-based Decomposition Algorithm for Nonlinear Mixed-Integer Optimal Control Problems

Adrian Bürger[a,b], Clemens Zeile[c], Angelika Altmann-Dieses[a], Sebastian Sager[c], and Moritz Diehl[b,d]

[a]Institute of Refrigeration, Air-Conditioning, and Environmental Engineering (IKKU), Karlsruhe University of Applied Sciences, Moltkestraße 30, 76133 Karlsruhe, Germany.

[b]Systems Control and Optimization Laboratory, Department of Microsystems Engineering (IMTEK), University of Freiburg, Georges-Koehler-Allee 102, 79110 Freiburg im Breisgau, Germany.

[c]Institute for Mathematical Optimization, Faculty of Mathematics, Otto von Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany.

[d]Department of Mathematics, University of Freiburg, Ernst-Zermelo-Straße 1, 79104 Freiburg im Breisgau, Germany.

September 29, 2022

In the following, we describe the minimum working example provided at `https://github.com/adbuerger/gn-miqp-mwe` and illustrate potential advantages of the Gauss-Newton-based decomposition approach presented in [1] for a numerical case study of Mixed-Integer Optimal Control (MIOC) of a simple nonlinear and unstable system cf. [2], Example 8.17, pp. 577-579. We describe the problem setup and implementation and compare the results of the proposed approach to results obtained using the Combinatorial Integral Approximation (CIA) problem [3].

Contents

1 Problem setup

We consider a simple Mixed-Integer Optimal Control Problem (MIOCP) of the form of Eq. (10) from [1] for a nonlinear unstable system with one state $x \in \mathbb{R}$ and one binary control $b \in \{0, 1\}$, cf. [2], Example 8.17, pp. 577-579. The continuous time system is described by

$$\dot{x} = x^3 - b \tag{1}$$

and transformed to a discrete time system

$$x^+ = f(x, b) \tag{2}$$

using one Runge-Kutta (RK)-4 step of step length $h = 0.05$. The aim is to track a reference $x_{\text{ref}} = 0.7$ starting from the initial value $x_0 = 0.8$ on a horizon of length $N = 30$, resulting in the following Mixed-Integer Non-Linear Program (MINLP):

$$\min_{\mathbf{x},\mathbf{b}} \; \Phi(\mathbf{x}) = \frac{1}{2}\sum_{k=0}^{N}(x(k) - x_{\text{ref}})^2 \tag{3a}$$

$$\text{s.t. } x(k+1) = f(x(k), b(k)), \quad k = 0, \dots, N-1, \tag{3b}$$

$$\mathbf{b} \in \mathbf{B} \cap \mathbb{Z}^N. \tag{3c}$$

$$x(0) = x_0, \tag{3d}$$

The combinatorial constraint set $\mathbf{B}$ imposes a minimum uptime constraint that requires that b remains active for at least three consecutive time steps, i.e., we have

$$\begin{aligned}\mathbf{B} = \{\mathbf{b} \in [0,1]^N \mid \\ & b(k) \geq b(k-1) - b(k-2), \\ & b(k) \geq b(k-1) - b(k-3), \quad k = 0, \dots, N-1\}.\end{aligned} \tag{4}$$

The required previous values $b(-1)$, $b(-2)$, $b(-3)$ are all set to zero. Further details on minimum dwell time constraints in the MIOCP context can be found in [4].

The described problem is solved by applying the decomposition algorithm presented in [5] once using the Combinatorial Integral Approximation (CIA)-Mixed-Integer Linear Program (MILP) [3] and once using the Gauss-Newton (GN)-Mixed-Integer Quadratic Program (MIQP). A description of the procedure and the individual steps S1, S2, and S3 is also given in [1].

2 Implementation

The MINLP is implemented using CasADi [6], which provides the possibility for the simplified generation of the derivatives of the MINLP components required for solving the Non-Linear Program (NLP) and setting up the GN-MIQP. The NLP stage in step S1 of the decomposition algorithm is solved using Ipopt [7]. Like in [8], we neglect

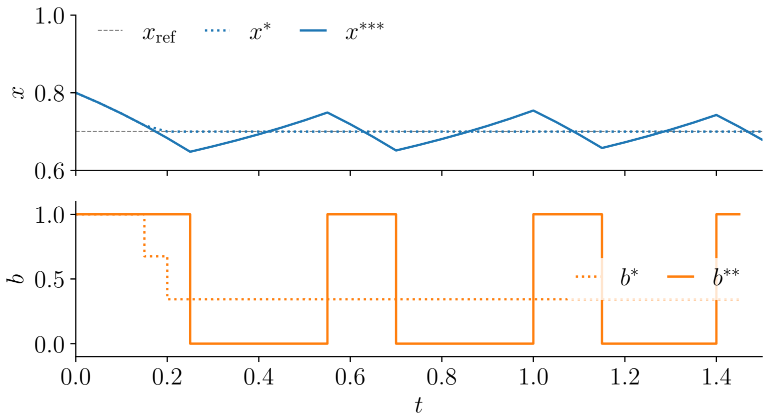

Figure 1: Relaxed and binary feasible solution for the GN-MIQP approach.

combinatorial constraints in step S1. The CIA problem is solved using a tailored branch-and-bound algorithm available in pycombina [9], the GN-MIQP is solved using Gurobi [10]. Owing to the absence of continuous controls, step S3 for this example just amounts to a system simulation. The code can be found in the file `miocp_decomposition.py`.

Moreover, the problem is solved for comparison using a branch-and-bound style simulation procedure described in Appendix A which has been implemented in Python and allows to determine the globally optimal solution. The code can be found in the file `miocp_branch_and_bound.py`.

3 Numerical results

Table 1 lists the objective values for (3) obtained using the described solution approaches and the objective value of the relaxed problem after step S1. It shows that for this setup, the GN-MIQP approach was able to achieve an improved solution compared with the CIA approach in terms of the MINLP objective. The solution obtained by the GN-MIQP approach here even is a globally optimal solution as the obtained objective value corresponds to the optimal solution obtained by the branch-and-bound style simulation.

The optimized state and control obtained by the GN-MIQP and CIA approach are given in Fig. 1 and Fig. 2, respectively. The reference value x_{ref} of the state is indicated

Table 1: Objective values of (3) for different solution approaches.

Solution approach	Objective value
Relaxed MINLP	$\Phi(\mathbf{x}^*) = 8.97 \cdot 10^{-3}$
Branch-and-bound style simulation (optimal)	$\Phi(\mathbf{x}^\circ) = 2.07 \cdot 10^{-2}$
CIA	$\Phi(\mathbf{x}_{\text{CIA}}^{***}) = 1.32 \cdot 10^{-1}$
GN-MIQP	$\Phi(\mathbf{x}_{\text{GN}}^{***}) = 2.07 \cdot 10^{-2}$

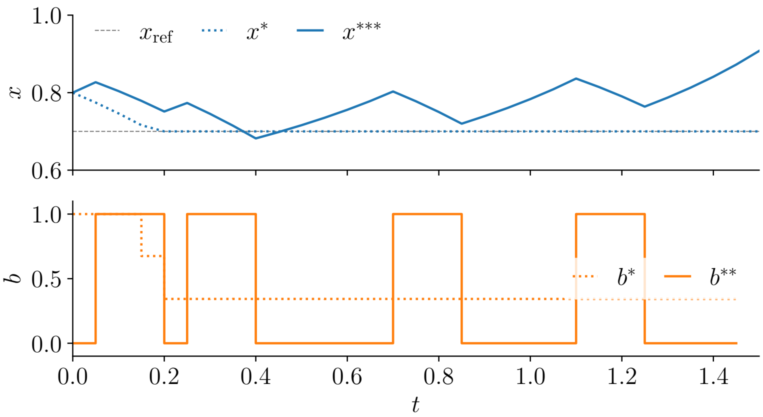

Figure 2: Relaxed and binary feasible solution for the CIA approach.

by a dashed grey line, the (identical) relaxed solutions (x^*, b^*) after step S1 by dotted lines, and the integer solutions $(x^{***}_{\mathrm{GN}}, b^{**}_{\mathrm{GN}})$ and $(x^{***}_{\mathrm{CIA}}, b^{**}_{\mathrm{CIA}})$ after step S3 by solid lines. For the relaxed solution, it can be observed that higher control effort is applied at the beginning of the control horizon to drive the system state towards the reference value, and afterward the state reference value is perfectly tracked by constantly applying a control input of $b^*_k \approx 0.34$, $k = 4, \ldots, N-1$.

By comparing the state and control trajectories shown in Fig. 1 and Fig. 2, the influence of the obtained integer approximation on the state trajectory development can be observed. While the integer solution obtained using the CIA approach is a globally optimal solution of the approximation problem in terms of the objective function of the CIA problem, an application of b^{**}_{CIA} for control of the unstable system causes the system state to diverge from its reference x_{ref}. In contrast, the integer solution obtained using the GN-MIQP approach can maintain the state in a region around the reference value.

A Branch-and-bound style simulation procedure for globally optimal solution of (3)

To obtain a globally optimal solution of the MINLP (3), we can consider the fact that there exists only one binary control b per discrete time interval but no other optimization variables, e. g., continuous controls or slack variables. As the only decision that can be made per discrete time interval is whether the binary control b is either 0 or 1, the state trajectory resulting from a given sequence of binary controls $\mathbf{b}$ can be obtained via forward simulation. Moreover, the objective function (3a) when evaluated up to a discrete time point j as in

$$\Phi_j(\mathbf{x}) = \frac{1}{2} \sum_{k=0}^{j} (x(k) - x_{\mathrm{ref}})^2 \tag{5}$$

with $j \in \{0, \dots, N\}$ yields a lower bound for all objective values $\Phi_l(\mathbf{x})$ with $l > j$, so that

$$\Phi_j(\mathbf{x}) \leq \Phi_l(\mathbf{x}). \tag{6}$$

These insights motivate the simulation procedure depicted in Algorithm 1. This procedure, which has been used in [2], allows us to compute the globally optimal solution of MINLP (3) and obtain the optimal trajectories of binary controls $\mathbf{b}^*$ and states $\mathbf{x}^*$ and the optimal objective value Φ^* . Each node $n = [n_\mathrm{p}^\circ, j, b^\circ, x^\circ, \Phi^\circ]$ used within this branch-and-bound style procedure contains information on the parent node n_p° (while root nodes reference empty nodes as parent nodes, as denoted by $\emptyset$), corresponding discrete time point j, applied binary control b°, state x°, and objective value Φ° at the discrete time point. Parameter N_mut is the number of consecutive time steps that a binary control must remain active after activation, with regard to the minimum uptime constraints (4).

References

[1] A Bürger, C Zeile, A Altmann-Dieses, S Sager, and M Diehl. A Gauss-Newton-based decomposition algorithm for nonlinear mixed-integer optimal control problems. *Submitted*, 2022. Available at `https://optimization-online.org/2022/04/8890/`.

[2] J B Rawlings, D Q Mayne, and M M Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill, 2nd edition, 2020. 3rd printing. Available at `https://sites.engineering.ucsb.edu/~jbraw/mpc/`.

[3] S Sager, M Jung, and C Kirches. Combinatorial integral approximation. *Mathematical Methods of Operations Research*, 73(3):363, 2011.

[4] C Zeile, N Robuschi, and S Sager. Mixed-integer optimal control under minimum dwell time constraints. *Mathematical Programming*, 188:653–694, 2021.

[5] S Sager. Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control. *Journal of Process Control*, 19(8):1238–1247, 2009.

[6] J A E Andersson, J Gillis, G Horn, J B Rawlings, and M Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[7] A Wächter and L T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[8] A Bürger, D Bull, P Sawant, M Bohlayer, A Klotz, D Beschütz, A Altmann-Dieses, M Braun, and M Diehl. Experimental operation of a solar-driven climate system with thermal energy storages using mixed-integer nonlinear model predictive control. *Optimal Control Applications and Methods*, 42(5):1293–1319, 2021.

[9] A Bürger, C Zeile, M Hahn, A Altmann-Dieses, S Sager, and M Diehl. pycombina: An open-source tool for solving combinatorial approximation problems arising in mixed-integer optimal control. In *IFAC-PapersOnLine*, volume 53, pages 6502–6508, 2020. 21st IFAC World Congress.

[10] Gurobi Optimization, LLC. Gurobi optimizer reference manual. `https://www.gurobi.com`, 2022. Last accessed September 18, 2022.

Algorithm 1: Branch-and-bound style simulation procedure

```
Initialize an empty queue Q ;
Initialize best node n* ← ∅ and upper bound Φ* ← ∞ ;
Initialize i ← 0, j ← 0, n° ← ∅, j° ← 0, b° ← 0, x° ← x_0, Φ° ← (x_0 − x_ref)^2 ;
foreach b ∈ {0, 1} do
    j ← j° ; i ← N_mut − 1 ;
    if (b° ≠ b) ∧ (b = 1) then
        // Consider minimum uptimes
        i ← 0 ;
    while i < N_mut do
        x° ← f(x°, b) ;
        Φ° ← Φ° + (x° − x_ref)^2 ;
        i ← i + 1 ; j ← j + 1 ;
    n ← [n°, j, b, x°, Φ°] ;
    Add n to Q ;
while Q is not empty do
    n° ← next node selected from Q with the highest depth j; if multiple nodes
      with the same depth j exist, select the one with the highest objective value
      Φ° ;
    Unpack [n°_p, j°, b°, x°, Φ°] ← n°;
    if j° = N then
        if Φ° < Φ* then
            // Solution update
            n* ← n° ; Φ* ← Φ° ;
    else
        foreach b ∈ {0, 1} do
            j ← j° ; i ← N_mut − 1 ;
            if (b° ≠ b) ∧ (b = 1) then
                i ← 0 ;
            while i < N_mut do
                x° ← f(x°, b) ;
                Φ° ← Φ° + (x° − x_ref)^2 ;
                i ← i + 1 ; j ← j + 1 ;
            n ← [n°, j, b, x°, Φ°] ;
            Add n to Q ;
Reconstruct b*, x* starting from n* by recursively following the parent node
  references until a root node is reached ;
return b*, x*, Φ*
```