Robust model predictive control exploiting monotonicity properties - Supplementary material

Moritz Heinlein[1], Sankaranarayanan Subramanian[2] and Sergio Lucia[1]

I. Decomposition Function

The system is decomposed into monotonically in x and p increasing parts which should correspond to $\overline{x}, \overline{p}$, and monotonically decreasing parts, corresponding to $\underline{x}, \underline{p}$. So, the minimum/maximum in each equation of (10) was under-/over-approximated by replacing each state in each individual term with $\overline{x}$ if this term is monotonically increasing in the respective state or with $\underline{x}$ if it is monotonically decreasing following Proposition 3. There is one exception to this procedure, as the state whose dynamic behaviour is determined by the respective equation always gets replaced by $\overline{x}$ (see [1]). The hyperrectangluar reachable sets for the CSTR case-study are computed according to following decomposition function:

$$d(\overline{x}, \overline{p}, u, \underline{x}, \underline{p}) = \left[d_{\dot{c}_A}, d_{\dot{c}_B}, d_{\dot{c}_R}, d_{\dot{c}_S}, d_{\dot{T}_R}\right]^\mathsf{T}, \tag{1a}$$

$$\begin{aligned} d_{\dot{c}_{A,i}} =& \frac{\dot{V}_\text{out}}{V_i}(\overline{c}_{A,i-1} - \overline{c}_{A,i}) + \frac{u_{A,i}}{V_i} \\ &- \underline{k}_{1,i} \exp\left(-\frac{E_{A1,i}}{R_\text{gas}(\underline{T}_{R,i} + 273.15)}\right)\overline{c}_{A,i}\underline{c}_{B,i} \\ &- 2\underline{k}_{2,i} \exp\left(-\frac{E_{A2,i}}{R_\text{gas}(\underline{T}_{R,i} + 273.15)}\right)\overline{c}_{A,i}{}^2, \end{aligned} \tag{1b}$$

$$\begin{aligned} d_{\dot{c}_{B,i}} =& \frac{\dot{V}_\text{out}}{V_i}(\overline{c}_{B,i-1} - \overline{c}_{B,i}) + \frac{u_{B,i}}{V_i} \\ &- \underline{k}_{1,i} \exp\left(-\frac{E_{A1,i}}{R_\text{gas}(\underline{T}_{R,i} + 273.15)}\right)\underline{c}_{A,i}\overline{c}_{B,i}, \end{aligned} \tag{1c}$$

$$\begin{aligned} d_{\dot{c}_{R,i}} =& \frac{\dot{V}_\text{out}}{V_i}(\overline{c}_{R,i-1} - \overline{c}_{R,i}) \\ &+ \overline{k}_{1,i} \exp\left(-\frac{E_{A1,i}}{R_\text{gas}(\overline{T}_{R,i} + 273.15)}\right)\overline{c}_{A,i}\overline{c}_{B,i}, \end{aligned} \tag{1d}$$

$$\begin{aligned} d_{\dot{c}_{S,i}} =& \frac{\dot{V}_\text{out}}{V_i}(\overline{c}_{S,i-1} - \overline{c}_{S,i}) \\ &+ \overline{k}_{2,i} \exp\left(-\frac{E_{A2,i}}{R_\text{gas}(\overline{T}_{R,i} + 273.15)}\right)\overline{c}_{A,i}{}^2, \end{aligned} \tag{1e}$$

$$\begin{aligned} d_{\dot{T}_{R,i}} =& \frac{\dot{V}_\text{out}}{V_i}(\overline{T}_{R,i-1} - \overline{T}_{R,i}) \\ &- \overline{k}_{1,i} \exp\left(-\frac{E_{A1,i}}{R_\text{gas}(\overline{T}_{R,i} + 273.15)}\right)\overline{c}_{A,i}\overline{c}_{B,i}\frac{\Delta H_{R1,i}}{\rho c_p} \\ &- \overline{k}_{2,i} \exp\left(-\frac{E_{A2,i}}{R_\text{gas}(\overline{T}_{R,i} + 273.15)}\right)\overline{c}_{A,i}{}^2\frac{\Delta H_{R2,i}}{\rho c_p} \\ &+ \frac{kA}{\rho c_p V_i}(T_{J,i} - \overline{T}_{R,i}). \end{aligned} \tag{1f}$$

II. Dividing the reachable sets

In this section, the mathematical constraints for the function h are defined for both the grid-like cutting and the search-tree like cutting.

*The research leading to these results has received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant agreement number 423857295. The authors thank M. Diehl, F. Messerer, M. Molnar and J. Köhler for fruitful discussions.

[1] M. Heinlein and S. Lucia are with the Chair of Process Automation Systems at Technische Universität Dortmund, Emil-Figge-Str. 70, 44227 Dortmund, Germany.

[2] S. Subramanian is with the Chair of Process Dynamics and Operations, Technische Universität Dortmund, Emil-Figge-Str. 70, 44227 Dortmund, Germany.

(a) Grid-like partitions (b) Search-tree-like partitions

Fig. 1: Methods on dividing the reachable set

Regardless of the method, each subregion needs to fulfill following constraint

$$x^{s-} - x^{s+} \leq 0, \forall s \in \mathbb{S}, \tag{2}$$

as the top right corners needs to lie above their respective bottom left corners.

A. Grid-like cutting

In [2], the dimensions normal to which the reachable sets are divided are predefined and the whole reachable set is divided along this cut.

In Figure 1a each division divides the whole reachable set like a mesh, leading to subregions, which are aligned in a grid. We define the vector $n_c \in \mathbb{Z}_{0+}^{n_x}$ as a vector of non-negative integers, stating the number of divisions per dimension, the respective cuts are normal to. In Figure 1a, the reachable set was divided twice normal to the first (horizontal) dimension and once normal to the second dimension, resulting in $n_c = [2,\ 1]^\intercal$. The total number of subregions can then be calculated as

$$\mu_s = \prod_{i=1}^{n_x} (n_{c,i} + 1).$$

As the cuts divide the whole reachable set, all the subregions affected by one cut are aligned in the dimension the cut is normal to. The subsets of all indices s counting these subregions which correspond to subsets aligned along one cut are denoted as $\mathbb{S}_i^m$, where i is the dimension, the cut is normal to, and m, where $0 \leq m < n_{c,i}$, is the number of cuts previous to this cut in the dimension i. Then one can write a subset of constraints defined in h $\forall i = 1, ..., n_x,\ \forall m = 0, ..., n_{c,i} - 1$

$$0 \leq x_i^{s\pm} - x_i^{s_0^{i,m}\pm} \leq 0,\ \forall s \in \mathbb{S}_i^m \backslash s_0^{i,m}, \tag{3}$$

where $s_0^{i,m}$ is the first element of $\mathbb{S}_i^m$.

Additionally, the top right corners of the subregions lying directly below one cut are related to the bottom left corners of the subregions directly above in the element of the dimension the cut is normal to. This leads to following equality constraints inside of h $\forall i = 1, ..., n_x$ and $\forall m = 0, ..., n_{c,i} - 2$, if $n_{c,i} > 0$

$$0 \leq x_i^{s_0^{i,m}+} - x_i^{s_0^{i,m+1}-} \leq 0. \tag{4}$$

Finally, the position of the cuts needs to be constrained to remain inside the reachable set. Therefore, the last inequality constraints in h are $\forall i = 1, ..., n_x$ and $\forall m = 0, ..., n_{c,i} - 2$, if $n_{c,i} > 0$

$$x_i^{1-} - x_i^{s_0^{i,m}+} \leq 0, \tag{5a}$$

$$x_i^{s_0^{i,m}+} - x_i^{\mu_s+} \leq 0. \tag{5b}$$

Concluding, we can write for the grid-like cutting

$$h_{\text{grid}}(x^{[1:\mu_s]\pm}) = [(2),\ (3),\ (4),\ (5)]\,,$$

as the concatenation of the left-hand sides of the referenced inequalities.

B. Search-tree-like cutting

This formulation has less constraints than (3)-(5) in Section II-A, but requires more parameters to be tuned.

Figure 1b shows the difference to the grid-like cutting. In contrast to the previous approach, after the reachable set has been divided once, further divisions in the resulting subregions can be made independently.

The ordering of the dimensions normal to which the subregions are cut is now important as the later the cuts are made, the more independent they can be made from each other. This can be represented by a vector $a \in \mathbb{N}^{n_x}$, which contains the ordering of the dimensions. The example in Figure 1b leads to $a = [1, 2]$, because first the reachable set is divided normal to dimension 1, before the resulting subregions are divided normal to dimension 2. It is also possible to have a different number of cuts normal to the same dimension in two already separated subregions. Therefore, as different numbers of cuts are allowed inside of regions, which have been divided already in another dimension (see Fig. 1b), μ_s cannot be calculated as before. Additionally, the index m for the sets $\mathbb{S}_i^m$ now cannot be associated directly with the number of cuts, as the cuts do not divide

the whole reachable set necessarily. However, these sets $\mathbb{S}_i^m$ still contain the indices s for each subregion, which is aligned in dimension i, however, the upper bound on m, k_i, grows larger as the number of different alignments in dimension i increases. Therefore, (3) now has to hold $\forall i = 1, ..., n_x$ and $\forall m = 0, ..., k_i - 1$.
The constraints connecting the adjacent subregions $\forall i = 1, ..., n_x$ and $\forall m = 0, ..., k_i - 1$ are

$$0 \leq x_{a_i}^{s_0^{a_i,m}+} - x_{a_i}^{s_0^{a_i,m+1}-} \leq 0, \qquad \text{if } \exists q : \{s_0^{a_i,m} s_0^{a_i,m+1}\} \subseteq \mathbb{S}_{a_{i-1}}^q, \tag{6}$$

as those adjacent subregions lie in the same region that was defined through the cut in the previous dimension. $s_0^{a_i,m}$ is defined as the first element in $\mathbb{S}_{a_i}^m$. $\mathbb{S}_{a_0}^0$ is defined to be equal to $\mathbb{S}$. Additionally the bottom left corners of all the subregions starting at the edge of the reachable set need to be constrained to lie on this edge, so $\forall i = 1, ..., n_x$ and $\forall m = 0, ..., k_i$

$$0 \leq x_{a_i}^{s-} - x_{a_i}^{1-} \leq 0, \ \forall s \in \mathbb{S}_{a_i}^m \text{ if } \exists q : s = s_0^{a_{i-1},q}. \tag{7}$$

Similarly for the top right corners must hold $\forall i = 1, ..., n_x$ and $\forall m = 0, ..., k_i$

$$0 \leq x_{a_i}^{s+} - x_{a_i}^{\mu_s+} \leq 0, \ \forall s \in \mathbb{S}_{a_i}^m \text{ if } \exists q : s = s_{-1}^{a_{i-1},q}, \tag{8}$$

where $s_{-1}^{i,m}$ denotes the last element in $\mathbb{S}_i^m$.
Summarizing, h_{tree} can be written analogously to h_{grid} as follows:

$$h_{\text{tree}} = [(2), (3), (6), (7), (8)] \, . \tag{9}$$

The number of branches of the tree can also be represented by a nested list l, whose number of elements are equal to the number of subregions divided in the dimension named in the first element of a. In each entry is a list with the number of branches corresponding to the dimension specified in the second element of a. For example, the list can look like

$$\begin{gathered} l = [[[[1, 2]] \, , [[3] \, , [4, 5]] \, , [[6] \, , [7, 8]]] \, , \\ [[[9] \, , [10, 11, 12]] \, , [[13, 14] \, , [15, 16]]]] \end{gathered}$$

The function len(l) gives the number of entries in the list l. The function depth(l) returns the number of levels of lists inside l. The function get_num(l) returns a vector with all numbers in the list, regardless of the nested structure. Then the elements of h can be determined by following recursive algorithm.

References

[1] M. Abate, M. Dutreix, and S. Coogan, “Tight Decomposition Functions for Continuous-Time Mixed-Monotone Systems With Disturbances,” *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 139–144, Jan. 2021.

[2] M. Heinlein, S. Subramanian, M. Molnar, and S. Lucia, “Robust MPC approaches for monotone systems*,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Dec. 2022, pp. 2354–2360.

Algorithm 1 Algorithm for Search-tree-like cutting

```
function CONSTR_FUNC(l, a, h, x_k^{[1:μ_s]±})
    for k = 1 : len(l) do
        idx = get_num(l[k])
        dim= a[−depth(l)]
        for s = idx do
            if s = idx[1] & k = 1 then
                h =concatenate(h, x_dim^{s−} − x_dim^{1−})
                h =concatenate(h, x_dim^{1−} − x_dim^{s−})
            else
                h =concatenate(h, x_dim^{s−} − x_dim^{idx[1]−})
                h =concatenate(h, x_dim^{idx[1]−} − x_dim^{s−})
            end if
            if s = idx[−1] & k = len(l) then
                h =concatenate(h, x_dim^{s+} − x_dim^{μ_s+})
                h =concatenate(h, x_dim^{μ_s+} − x_dim^{s+})
            else
                h =concatenate(h, x_dim^{s+} − x_dim^{idx[−1]−})
                h =concatenate(h, x_dim^{idx[−1]−} − x_dim^{s+})
            end if
        end for
        if k > 1 then
            prev_last=get_num(l[k − 1])[−1]
            h =concatenate(h, x_dim^{idx[1]−} − x_dim^{prev_last+})
            h =concatenate(h, x_dim^{prev_last+} − x_dim^{idx[1]−})
        end if
        if depth(l) > 1 then
            h =constr_func(l[k], a, h, x_k^{[1:μ_s]±})
        end if
    end for
    return h
end function
h = [ ]
for s = 1 : μ_s do
    h =concatenate(h, x^{1−} − x^{s±})
    h =concatenate(h, x^{s±} − x^{μ_s+})
end for
for i=1:N do
    h =constr_func(l, a, h, x_i^{[1:μ_s]±})
end for
```