Model predictive control with memory-based discrete search for switched linear systems

Larsen, Rie B.; Atasoy, Bilge; Negenborn, Rudy R.

DOI
10.1016/j.ifacol.2020.12.325

Publication date
2020

Document Version
Final published version

Published in
IFAC-PapersOnline

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-2 (2020) 6769–6774

Model Predictive Control with Memory-based Discrete Search for Switched Linear Systems ⋆

Rie B. Larsen * **Bilge Atasoy** * **Rudy R. Negenborn** *

* *Department of Maritime and Transport Technology, Delft University of Technology. The Netherlands (e-mail: r.b.larsen|b.atasoy|r.r.negenborn@tudelft.nl).*

Abstract: Controlling systems with both continuous and discrete actuators using model predictive control is often impractical, since mixed-integer optimization problems are too complex to solve sufficiently fast. This paper proposes a parallelizable method to control both the continuous input and the discrete switching signal for linear switched systems. The method uses ideas from Bayesian optimization to limit the computation to a predefined number of convex optimization problems. The recursive feasibility and stability of the method is guaranteed for initially feasible solutions. Results from simulated experiments show promising performances and computation times.

Keywords: predictive control, integer control, parallel computation, computational methods, stability analysis, mixed-integer optimization, rolling horizon

1. INTRODUCTION

Many systems are controlled by a combination of discrete and continuous actuators. A class of them can be described by switched linear dynamics without dwell-time where both the continuous input and the discrete switching signals are decisions taken by the applied controller. For very small systems, model predictive control (MPC) can be applied directly to improve the performance of these systems under constraints. However, since the system must be described using both continuous and discrete variables, the MPC controller must solve a mixed-integer program every time the control sequence is updated. For larger systems, the computation of this mixed-integer program may be unacceptably long compared to the system specifications.

Switched linear systems can be categorized as either internally or externally forced. An overview of stability analysis for different classes of switched linear systems can be found in Lin and Antsaklis (2009), while Zhu and Antsaklis (2015) reviews the literature on MPC for switched systems. The literature on MPC for internally forced systems focuses mainly on the case where the switching signal is state dependent (piecewise affine systems) as in, e.g., Morari and Bari (2006), or on systems under uncontrolled switching, as in, e.g., Zhang et al. (2016). This paper considers externally forced systems, hence systems where the applied MPC must decide on both the continuous inputs and the discrete switching signals.

To improve the computation time of MPC for externally forced switched linear systems, most frameworks focus on solving each iteration faster. Often the proposed integer solvers are iteration based, e.g., Axehill et al. (2014) or Naik and Bemporad (2017), which limits how fast the computation can be and increases the computation time uncertainty. This complicates the implementation of MPC, since a typical MPC re-computes the input at predefined time intervals. One way to avoid performing iterative computations of the discrete variables is using convex relaxations of the mixed-integer problem. In Sager et al. (2012) it is shown that continuous variables can be used to approximate binary variables in the MPC optimization problem if the timesteps are sufficiently small. In Gau et al. (2017) this result is used to control a switch linear system without other input than the switching signal. In Prada et al. (2009) the switching signals are modelled as switching times resulting in a non-linear optimization problem with solely continuous variables. Mendes et al. (2017) decrease the number of iterations needed to solve the mixed-integer problem by distributing the computation between different agents coupled by auxiliary variables.

Another stream of research, namely that on suboptimal MPC, acknowledge that it is not always possible to obtain the optimal solution. This research establishes bounds and criteria on the suboptimality and the properties of the implemented control law that ensures stability and feasibility of the system. Stability and recursive feasibility of the suboptimal MPC is in, e.g., Lazar and Heemels (2009) and Pannocchia et al. (2011) established by imposing specific improvements in the objective function. In contrast, Scokaert et al. (1999) ensure stability and recursive feasibility by improving a known solution which satisfy stability and feasibility criteria.

In this paper, the proposed method finds better solutions by remembering the performance at previous timesteps,

⋆ This research is supported by the project "Complexity Methods for Predictive Synchromodality" (project 439.16.120) of the Netherlands Organisation for Scientific Research (NWO).

10.1016/j.ifacol.2020.12.325

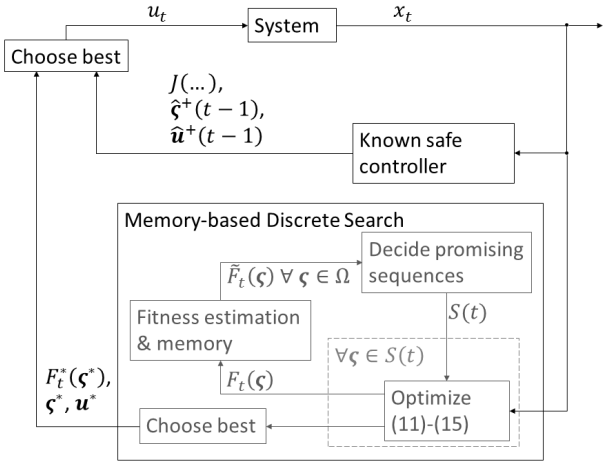

Fig. 1. Schematics of MPC with MDS.

and is thus called MPC with Memory-based Discrete search (MDS). The method is outlined in Figure 1 with notation as introduced in the remainder of this paper. MPC with MDS solves the part of the usual MPC problem, which is continuous in the variables, for a set of switching signal sequences. It then implements the first elements of the sequence with the best performance, if this performance is better than that of a stabilizing input sequence computed based on the previous MPC prediction and a known conservative control law . If the known stabilizing input sequence performs better, its solution corresponding to the current timestep is implemented and the process starts over.

It is assumed that the performance of a switching signal sequence at a given time is comparable to the performance of related sequences at the previous timestep. This assumption is used to choose the switching signal sequences for which MPC with MDS computes the convex optimization problem. The ideas are analogous to Bayesian Optimization where the current knowledge of a function is used to assess which function evaluation to perform at the next iteration. See e.g. Jones et al. (1998) or Snoek et al. (2012) for an introduction. MPC with MDS utilizes the rolling horizon from MPC to obtain information about not only current solutions, but also future solutions. In this fashion, the optimization problems in MPC with MDS has low complexity and can be solved in parallel.

The remainder of the paper is organized as follows: in Section 2, the considered system and problem is specified. In Section 3, MPC with MDS is presented together with an analysis of stability and recursive feasibility. In Section 4, MPC with MDS is assessed using a simulation and compared with three other controllers. Finally, Section 5 draws the main conclusions and outlines future research.

2. SYSTEM AND PROBLEM DEFINITION

We consider a discrete time, switched, linear system without dwell-time with the following dynamics:

$$x_{t+1} = A_{\sigma_t} x_t + B_{\sigma_t} u_t, \tag{1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^{m_{\sigma_t}}$ are the state of and the input applied to the system at discrete time t. The switching signal $\sigma_t \in \mathbb{S} = \mathbb{I}_{[1:l]}$ is the input at time t, that determines from finite sets, the dynamic matrices $A_{\sigma_t} \in \mathcal{A} = \{A_1, A_2, ..., A_l\}$ and $B_{\sigma_t} \in \mathcal{B} = \{B_1, B_2, ..., B_l\}$. $\mathbb{I}_{[a;b]}$ denotes the set of integers $\{x \in \mathbb{I} | a \leq x \leq b\}$. The system is controllable for one or more switching signals.

Definition 1. The basic switching signal is denoted by γ. It is one switching signal $\gamma \in \mathbb{S}$ for which system (1) is controllable. The corresponding dynamics $x_{t+1} = A_\gamma x_t + B_\gamma u_t$ will be referred to as the basic dynamics.

The system's state and inputs are constrained by

$$x_t \in \mathbb{X} \subseteq \mathbb{R}^n \qquad \forall\, t \tag{2}$$
$$u_t \in \mathbb{U}_{\sigma_t} \subseteq \mathbb{R}^{m_{\sigma_t}} \forall\, t. \tag{3}$$

The set $\mathbb{X}$ is convex, closed and contains the origin in its interior, while each set $\mathbb{U}_{\sigma_t}$ is convex and compact. $\mathbb{U}_\gamma$ contains the origin.

It is assumed that more information about the basic dynamics is known.

Assumption 1. A control law $u = \kappa(x)$, called the safe control law, and a set $\mathcal{X}_f$ are known for which the following is true:

(1) The basic dynamics under the safe control law, i.e. $x_{t+1} = A_\gamma x_t + B_\gamma \kappa(x_t)$ is asymptotically stable.
(2) $\mathcal{X}_f \subseteq \mathbb{X}$, $0 \in \mathcal{X}_f$ and $\mathcal{X}_f$ is closed.
(3) $\kappa(x) \in \mathbb{U}_\gamma\ \forall\, x \in \mathcal{X}_f$.
(4) $A_\gamma x + B_\gamma \kappa(x) \in \mathcal{X}_f\ \forall\, x \in \mathcal{X}_f$.

2.1 Problem definition

This paper considers stability to the equilibrium at the origin, but the results can be generalized to set point stability.

When system (1) is controlled by an MPC controller, the controller solves at timestep t the following mixed-integer program:

$$\min_{\substack{u_0, ..., u_{N-1} \\ \varsigma_0, ..., \varsigma_{N-1}}} \quad J(\boldsymbol{\varsigma}, \boldsymbol{u}, \boldsymbol{x}) \tag{4}$$

$$J(\boldsymbol{\varsigma}, \boldsymbol{u}, \boldsymbol{x}) = \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R_{\varsigma_k} u_k \right) + x_N^T Q_N x_N \tag{5}$$

$$\text{s.t} \quad x_0 = x_t \tag{6}$$
$$x_{k+1} = A_{\varsigma_k} x_k + B_{\varsigma_k} u_k \quad \forall\, k \in \mathbb{I}_{[0;N-1]} \tag{7}$$
$$x_k \in \mathbb{X},\ u_k \in \mathbb{U}_{\varsigma_k},\ \forall\, k \in \mathbb{I}_{[0;N-1]} \tag{8}$$
$$x_N \in \mathcal{X}_f, \tag{9}$$

where variables $x_k \in \mathbb{R}^n$, $\varsigma_k \in \mathbb{I}_{[1:l]}$ and $u_k \in \mathbb{R}^{m_{\varsigma_k}}$ represent the controller's model of the state, switching signal and input to the system at prediction time $t + k$, respectively. N is the prediction horizon. The cost matrices R and Q_N are symmetric and positive definite, while Q is symmetric and at least positive semi-definite. Bold symbols denote ordered time sequences, e.g., $\boldsymbol{a}_{[0;N]} = \langle a_0, a_1, ... a_N \rangle$, where each element a_k is a vector related to timestep k. When appropriate, the interval subscript is omitted for a simpler notation.

Assumption 2. The stage cost for the basic dynamics $l(x, u, \gamma) = x_k^T Q x_k + u_k^T R_\gamma u_k$ and the

Algorithm 1 MPC with MDS

Let $\hat{\boldsymbol{\varsigma}}(t)$, $\hat{\boldsymbol{u}}(t)$ and $\hat{\boldsymbol{x}}(t)$ be a feasible solution to (4)-(9)
while system is to be controlled **do**
  Implement $\sigma_t = \hat{\varsigma}_0$, $u_t = \hat{u}_0$
  $t = t + 1$
  Measure x_t
  Compute $\mathcal{S}(t)$ using Algorithm 2
  for all $\boldsymbol{\varsigma} \in \mathcal{S}(t)$ **do**
    if (10)-(14) is feasible **do**
      $\tilde{F}_t(\boldsymbol{\varsigma}) = F_t(\boldsymbol{\varsigma})$
    else
      $\tilde{F}_t(\boldsymbol{\varsigma}) = M$
    end if
    $\tilde{Y}_t(\boldsymbol{\varsigma}) = \beta J\left(\hat{\boldsymbol{\varsigma}}^+(t-1), \boldsymbol{u}^+(t-1), \boldsymbol{x}^+(t-1)\right)$
  end for
  Find $F_t^*(\boldsymbol{\varsigma}^*) = \min_{\boldsymbol{\varsigma} \in \mathcal{S}(t)} F_t(\boldsymbol{\varsigma})$
  if $F_t^*(\boldsymbol{\varsigma}^*) \leq J\left(\hat{\boldsymbol{\varsigma}}^+(t-1), \boldsymbol{u}^+(t-1), \boldsymbol{x}^+(t-1)\right)$ **do**
    $\hat{\boldsymbol{\varsigma}}(t) = \boldsymbol{\varsigma}^*$, $\hat{\boldsymbol{u}}(t) = \boldsymbol{u}^*$, $\hat{\boldsymbol{x}}(t) = \boldsymbol{x}^*$
  else
    $\hat{\boldsymbol{\varsigma}}(t) = \hat{\boldsymbol{\varsigma}}^+(t-1)$, $\hat{\boldsymbol{u}}(t) = \hat{\boldsymbol{u}}^+(t-1)$, $\hat{\boldsymbol{x}}(t) = \hat{\boldsymbol{x}}^+(t-1)$
  end if
end while

final cost $V_f(x) = x^T Q_N x$ are such that $V_f(A_\gamma x + B_\gamma \kappa(x)) - V_f(x) + l(x, \kappa(x), \gamma) \leq 0 \; \forall x \in \mathcal{X}_f$.

Notice that Assumption 1 and 2 can by satisfied by the design of $\kappa(x)$, $\mathcal{X}_f$ and $V_f(x)$.

The time it takes to solve (4)-(9) often limits which switched linear systems can realistically be controlled with MPC. Most solvers are iteration based, and thus not suitable for parallel implementations. Furthermore, current efforts typically do not take the recursive nature of MPC into account.

3. MEMORY-BASED DISCRETE SEARCH

Instead of solving a slow mixed-integer problem at each timestep, we suggest to search over the possible sequences of switching signals based on their previous performance and only solve the MPC problem for the chosen sequences. We call the method MPC with Memory-based Discrete Search (MDS). For given switching signal sequences, the optimization problem corresponding to each sequence is continuous in the variables, and hence solvable by fast, convex optimization solvers. When the set of switching signals, for which the MPC problem should be solved, depends on the method's memory of previous performances, the optimizations can be performed in parallel to decrease the computation time further.

MPC with MDS searches for good switching signal sequences based on previous performance. Since the controlled system's state changes dynamically, the expected performance of a given sequence will depend on the previous performance of sequences that are time-shifted and similar, not identical to that sequence. MPC with MDS is thus very suitable for systems that require long prediction horizons, as more information about a sequence's potential performance can be gathered before the dynamic information becomes obsolete.

To choose for which set of switching signal sequences to solve the MPC problem, MPC with MDS uses ideas from Bayesian optimization to ensure the set contains both sequences that are expected to perform well, and sequences that will provide new information. To measure the performance and information value of a switching signal sequence, the sequence's fitness $F_t(\boldsymbol{\varsigma}_{[0;N-1]})$, expected fitness $\tilde{F}_t(\boldsymbol{\varsigma}_{[0;N-1]})$ and uncertainty value $Y_t(\boldsymbol{\varsigma}_{[0;N-1]})$ are introduced.

Definition 2. The fitness of switching signal sequence $\boldsymbol{\varsigma}_{[0;N-1]} = \{\varsigma_0, ..., \varsigma_{N-1}\}$ is

$$F_t(\boldsymbol{\varsigma}_{[0;N-1]}) = \min_{u_1,\ldots,u_{N-1}} J(\boldsymbol{\varsigma}_{[0;N-1]}, \boldsymbol{u}, \boldsymbol{x}) \tag{10}$$

$$\text{s.t} \quad x_0 = x_t \tag{11}$$

$$x_{k+1} = A_{\varsigma_k} x_k + B_{\varsigma_k} u_k \quad \forall\, k \in \mathbb{I}_{[0;N-1]} \tag{12}$$

$$x_k \in \mathbb{X},\; u_k \in \mathbb{U}_{\varsigma_k},\; \forall\, k \in \mathbb{I}_{[0;N-1]} \tag{13}$$

$$x_N \in \mathcal{X}_f \tag{14}$$

where the notation corresponds to that of (4)-(9).

Definition 3. The expected fitness for a sequence $\boldsymbol{\varsigma}$ at time t is denoted by $\tilde{F}_t(\boldsymbol{\varsigma})$ and its uncertainty value is denoted by $\tilde{Y}_t(\boldsymbol{\varsigma})$.

Figure 1 presents a schematics of MPC with MDS. The detailed notation will be provided in the remainder of this section. At each timestep t, fitness estimates of all switching signal sequences are used to determine a set of promising sequences. For each promising sequence, the fitness is evaluated for the current system state. The best performance of the promising sequences is then compared to the performance of a known controller. This controller knows a switching signal sequence and a corresponding continuous input sequence which satisfy all constraints and stabilizes the system. The first switching signal and continuous input corresponding to the sequence that performs the best are implemented on the system and the process starts over at the next timestep.

How the promising sequences are chosen does not affect the stability of a system controlled by MPC with MDS. Therefore, the stability and recursive feasibility of the proposed method are proven in Section 3.1 before the selection method is presented in Section 3.2.

3.1 Stability and feasibility

To guarantee stability and recursive feasibility of the MPC problem, MPC with MDS compares the performance of the potential sequences with the performance of a control law, which is known to satisfy all constraints. This section outlines how this is done, and proves that the implemented inputs will stabilize the system.

The algorithm knows a feasible solution at time t, but needs to access the performance of the potential sequences at time $t+1$. It is thus necessary to define how the solution at time t is shifted in time and to prove that the shifted solution is indeed feasible.

Definition 4. The sequences $\boldsymbol{\varsigma}^+_{[0;N-1]} = \{\boldsymbol{\varsigma}_{[1;N-1]}, \gamma\}$, $\boldsymbol{u}^+_{[0;N-1]} = \{\boldsymbol{u}_{[1;N-1]}, \kappa(x_N)\}$ and $\boldsymbol{x}^+_{[0;N]} = \{\boldsymbol{x}_{[1;N]}, A_\gamma x_N + B_\gamma \kappa(x_N)\}$ are said to be the shifted sequences of $\boldsymbol{\varsigma}_{[0;N-1]}$, $\boldsymbol{u}_{[0;N-1]}$ and $\boldsymbol{x}_{[0;N]}$.

The outer framework of MPC with MDS can be seen in Algorithm 1. The method is initialized with a feasible solution to the mixed-integer problem (4)-(9) for the initial state. In the algorithm the best solution that is known to the controller at a given time is marked with a hat. The first elements of the best known switching signal sequence and continuous input sequence are implemented as is usually done with MPC. Hereafter MPC with MDS decides on the set of potential sequences. How the method computes $\mathcal{S}(t)$ in line 6 will be detailed in Section 3.2. For feasible sequences, the expected fitness is updated to be the actual fitness, while infeasible sequences' expected fitnesses are penalized with a large number M. The uncertainty value is in both cases updated to be a factor, $\beta > 0$, times the objective function value of the best known solution. This ensures that the uncertainty values are of reasonable size compared to the fitness values. If the best of the potential sequences is better than the best known solution, that potential sequence becomes the new best solution known by the controller.

Lemma 1. For any switching signal sequence $\hat{\boldsymbol{\varsigma}}_{[0;N-1]}$ that has a feasible solution to (10)-(14) at time t, the shifted sequence $\hat{\boldsymbol{\varsigma}}^{+}_{[0;N-1]}$ will have a feasible solution to (10)-(14) at time $t+1$.

Proof. Lemma 1 follows directly from constraint (14) and Assumption 1.

With the outer framework of MPC with MDS fully defined, we now analyse the stability of a system under Algorithm 1 using standard techniques, see, e.g., Mayne et al. (2000).

Theorem 2. If a feasible solution to (4)-(9) exists for System (1) at time $k = k_0$, then the system under Algorithm 1 converges to the origin.

Proof. To ensure stability, first recursive constraint satisfaction must be established. *Recursive Feasibility:* If (10)-(14) is infeasible at any time t for all $\boldsymbol{\varsigma} \in \mathcal{S}(t)$ the sequences from $t-1$ are shifted and applied by Algorithm 1. The recursive feasibility of the shifted sequences is given by Lemma 1. *Stability:* Sufficient criteria for stability of System (1) subject to Algorithm 1 are A) $J(\hat{\boldsymbol{\varsigma}}(t+1), \hat{\boldsymbol{u}}(t+1), \hat{\boldsymbol{x}}(t+1)) < J(\hat{\boldsymbol{\varsigma}}(t), \hat{\boldsymbol{u}}(t), \hat{\boldsymbol{x}}(t))$ $\forall \; t \in \{t | J(\hat{\boldsymbol{\varsigma}}(t), \hat{\boldsymbol{u}}(t), \hat{\boldsymbol{x}}(t)) > 0\}$ and B) $J(\hat{\boldsymbol{\varsigma}}(t+1), \hat{\boldsymbol{u}}(t+1), \hat{\boldsymbol{x}}(t+1)) = 0$ $\forall \; t \in \{t | J(\hat{\boldsymbol{\varsigma}}(t), \hat{\boldsymbol{u}}(t), \hat{\boldsymbol{x}}(t)) = 0\}$, because $J(\hat{\boldsymbol{\varsigma}}(t), \hat{\boldsymbol{u}}(t), \hat{\boldsymbol{x}}(t)) = 0$ only at the origin and $\sigma_t = \hat{\varsigma}_0$, $u_t = \hat{u}_0 \; \forall t$. Due to Line 16 in Algorithm 1 and Assumption 1, $J(\hat{\boldsymbol{\varsigma}}(t+1), \hat{\boldsymbol{u}}(t+1), \hat{\boldsymbol{x}}(t+1)) \leq J\left(\hat{\boldsymbol{\varsigma}}^{+}(t), \hat{\boldsymbol{u}}^{+}(t), \hat{\boldsymbol{x}}^{+}(t)\right) \; \forall t$. Using the notation of Assumption 2, criterion A) thus corresponds to $V_f(A_\gamma \hat{x}_N(t) + B_\gamma \kappa(\hat{x}_N(t))) - V_f(\hat{x}_N(t)) + l(\gamma, \kappa(\hat{x}_N(t)), \hat{x}_N(t)) - l(\hat{\varsigma}_0(t), \hat{u}_0(t), \hat{x}_0(t)) < 0$. The time-argument is omitted for the rest of the proof. Due to convexity $l(\hat{\varsigma}_0, \hat{u}_0, \hat{x}_0) > 0$ except at the origin, where $l(\hat{\varsigma}_0, \hat{u}_0, \hat{x}_0) = 0$. Both criteria A) and B) are thus fulfilled if $V_f(A_\gamma \hat{x}_N + B_\gamma \kappa(\hat{x}_N)) - V_f(\hat{x}_N) + l(\gamma, \kappa(\hat{x}_N), \hat{x}_N) \leq 0 \; \forall t$ which is ensured by Assumption 2.

Notice that the strategy used to update $\mathcal{S}(t)$ has no influence on the feasibility and stability properties of Algorithm 1. It only affects the quality of the solution and the computation time. Notice furthermore that infeasible solutions and suboptimal solutions to (10)-(14) do not impact the stability and feasibility of MPC with MDS.

Algorithm 2 Deciding $\mathcal{S}(t)$

1: **for** all $\boldsymbol{\varsigma} \in \boldsymbol{\Omega}$ **do**
2: $\quad \tilde{F}_t(\boldsymbol{\varsigma}) = \alpha \tilde{F}_{t-1}(\mathbf{a}(\boldsymbol{\varsigma})) + \frac{1-\alpha}{o_{\boldsymbol{\varsigma}}} \sum_{\boldsymbol{\psi} \in \mathcal{N}_{\boldsymbol{\varsigma}}} \tilde{F}_{t-1}(\boldsymbol{\psi})$
3: $\quad \tilde{Y}_t(\boldsymbol{\varsigma}) = \alpha \tilde{Y}_{t-1}(\mathbf{a}(\boldsymbol{\varsigma})) + \frac{1-\alpha}{o_{\boldsymbol{\varsigma}}} \sum_{\boldsymbol{\psi} \in \mathcal{N}_{\boldsymbol{\varsigma}}} \tilde{Y}_{t-1}(\boldsymbol{\psi})$
4: **end for**
5: $S(t) = \emptyset$
6: **while** $S(t)$ is not full **do**
7: $\quad S(t) = S(t) \cup \underset{\boldsymbol{\varsigma} \in \boldsymbol{\Omega} \setminus S(t)}{\arg\min} \tilde{F}_t(\boldsymbol{\varsigma})$
8: $\quad S(t) = S(t) \cup \underset{\boldsymbol{\varsigma} \in \boldsymbol{\Omega} \setminus S(t)}{\arg\min} \tilde{F}_t(\boldsymbol{\varsigma}) - \tilde{Y}_t(\boldsymbol{\varsigma})$
9: $\quad S(t) = S(t) \cup \underset{\boldsymbol{\varsigma} \in \boldsymbol{\Omega} \setminus S(t)}{\arg\max} \tilde{Y}_t(\boldsymbol{\varsigma})$
10: $\quad S(t) = S(t) \cup \{\boldsymbol{\psi}\}$ where $\boldsymbol{\psi} \in \boldsymbol{\Omega} \setminus S(t)$
11: **end while**
12: **return** $S(t)$

3.2 Sequence selection

Several methods can be used to select the set of potential sequences. MPC with MDS estimates, like in Bayesian optimization, whether a sequence is likely to be either the best solution (exploitation) or bring new information (exploration). The set of potential sequences is assembled to reflect a trade-off between exploitation and exploration. In the following it is assumed that N and size l of $\mathbb{S}$ are sufficiently small, that saving and sorting information on each l^N switching signal is realistic. If l and N are large, sampling methods and parallelization can be used to represent the switching signals.

When the fitness and uncertainty value of a sequence is estimated, only information from the previous timestep is used. It is thus necessary to establish what switching signal sequences from the previous time $t-1$ will have influence at a given sequence at time t. For this, we define neighbour sequences and implemented ancestors.

Definition 5. Two switching signal sequences $\boldsymbol{\varsigma}^a$ and $\boldsymbol{\varsigma}^b$ are neighbours, if they differ at only one timestep except for the last timestep, that is $\boldsymbol{\varsigma}^a \in \{\boldsymbol{\varsigma} | \varsigma_i = \varsigma_i^b \; \forall \, i \in \mathbb{I}_{[0;N-2]} \setminus \{j\}$, where $j \in \mathbb{I}_{[0;N-2]}\}$. Switching signal sequence $\boldsymbol{\varsigma}^a$ has $o_{\boldsymbol{\varsigma}^a}$ neighbours.

Definition 6. A switching sequence $\boldsymbol{\varsigma}$'s implemented ancestor $\mathbf{a}(\boldsymbol{\varsigma})$ at time t is a switching sequence whose first element is applied to the system at time $t-1$, and whose remaining elements are identical with the first $N-2$ elements in the sequence in question, i.e. switching signal sequence $\boldsymbol{\varsigma}$ has implemented ancestor $\mathbf{a}(\boldsymbol{\varsigma}) = \{\boldsymbol{\psi} | \sigma_{t-1} = \boldsymbol{\psi}_0 \, , \; \boldsymbol{\psi}_{[1;N-1]} = \boldsymbol{\varsigma}_{[0;N-2]}\}$.

A switching sequence's neighbours can be precomputed and does not vary over time, while the implemented ancestor will vary depending on the switching signal implemented at the previous timestep. However, a set of potential ancestors of a switching signal sequence can be precomputed and the correct implemented ancestor can be found online. This reduces computation time.

Algorithm 2 shows how MPC with MDS computes expected fitness and uncertainty value for the switching signal sequences and uses simple optimization over these values to decide the set of potential sequences. Before

Algorithm 1 is started, $\tilde{F}_t(\boldsymbol{\varsigma})$ and $\tilde{Y}_t(\boldsymbol{\varsigma})$ must be given initial values. We recommend a positive factor times the optimal function value of the initially known solution, to ensure correct scaling.

The expected fitness $\tilde{F}_t(\boldsymbol{\varsigma})$ and uncertainty value $\tilde{Y}_t(\boldsymbol{\varsigma})$ are updated for all sequences in $\boldsymbol{\Omega}$ at all times t. $\boldsymbol{\Omega}$ contains all possible switching signal sequences, even those for which (10)-(14) may be infeasible at time t. When the values are updated, the impact of the implemented ancestor is weighted to the impact of the average of the neighbour sequences by $0 < \alpha < 1$. Notice that the expected fitness and uncertainty value could be easily computed in parallel. The selection of the set of potential sequences could also be parallelized, if a centralized step ensures the uniqueness of each potential sequence in the set by replacing duplicates with random sequences. This will lead to increased exploration.

4. IMPLEMENTATION EXAMPLE

To illustrate the performance of the proposed method, simulation experiments are conducted where MPC with MBS has been implemented on a small switched linear system and the performance of MPC with MBS is compared to three benchmark controllers. In the following, first the benchmark algorithms are introduced, then the system is given, and finally the results are shown.

4.1 Benchmark controllers

The performance of MPC with MDS is compared to three other controllers, named Optimal, Random and Safe. All controllers are implemented in serial, since the authors do not have access to parallel computing yet. MPC with MDS and the Random controller are however prepared for parallel implementation as mentioned in the end of Section 3. The experiments are implemented in Matlab using Yalmip, Löfberg (2004), and Gurobi.

MPC with MDS: follows Algorithm 1 and 2 with the selection of the potential sequences prepared for parallelization as mentioned in the end of Section 3.2. (10)-(14) is solved using Gurobi's barrier method limited to 150 iterations. The experiment is repeated 10 times to illustrate the effect of the random selections used in Algorithm 2.

Optimal controller: solves the optimal mixed-integer MPC problem (4)-(9) at each time t. To obtain a convex continuous approximation of the mixed-integer program, the switching signals are represented by binary variables.

Safe controller: solves (4)-(9) for the system's initial state and uses the solution as switching signal and continuous input the first 9 timesteps. Hereafter the basic switching signal and the safe control law is implemented, i.e. $\sigma_t = \gamma$ and $u_t = \kappa(x_t)\ \forall\, t \geq 10$.

Random controller: selects the set of potential sequences randomly instead of using Algorithm 2 but follows otherwise Algorithm 1. (10)-(14) is solved using Gurobi's barrier method with a limitation of 150 iterations. The experiment is repeated 10 times.

4.2 System

The system used to illustrate the performance has initial state $x_0 = [-7\ -2]^T$ and dynamics

$$x_{t+1} = A_\sigma x_t + B_\sigma u_t \quad \text{where } \sigma \in [0,1,2,3],$$

$$A_0 = A_2 = \begin{bmatrix} 0.9 & 0.1 \\ 0 & 1.1 \end{bmatrix},\ B_0 = B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix},\ A_3 = \begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix} \text{ and } B_3 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

It is known that the unconstrained system is stable when $\sigma_t = 0$ and $u_t = [-0.1\ -0.2]\, x_t$ for all t.

The system is subject to state constraint $H_s x_t \leq h_s\ \forall\, t$, and input constraints $H_\sigma x_t \leq h_\sigma\ \forall\, t$, with

$$H_s = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0.1 & 0.2 \end{bmatrix}, h_s = \begin{bmatrix} 10 \\ 10 \\ 5 \\ 5 \\ 1 \end{bmatrix}, H_3 = \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, h_3 = \begin{bmatrix} 3 \\ -1 \\ 3 \\ -1 \\ 3 \end{bmatrix}$$

$$H_0 = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ h_0 = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix},\ H_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ h_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix},$$

$$H_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \text{ and } h_2 = \begin{bmatrix} 5 \\ -0.05 \end{bmatrix}.$$

Notice that the continuous input under switching signal 2 is bounded away from zero and that the continuous input under switching signal 4 has a different dimension.

Between timestep $t = 50$ and $t = 100$ the system is subject to a constant disturbance

$$x_{t+1} = A_\sigma x_t + B_\sigma \kappa(x_t) - \begin{bmatrix} 0.1 \\ 0 \end{bmatrix}\ \forall\, 50 \leq t \leq 98. \quad (15)$$

The controllers have no information about this disturbance, besides the measured state, and thus do not guarantee state constraint satisfaction. The disturbance is however so small that the Optimal controller remains feasible.

Both MPC with MDS, the Optimal and the Random controller are implemented with prediction horizon $N = 10$ and the following cost

$$J(\boldsymbol{\varsigma}, \boldsymbol{u}, \boldsymbol{x}) = \sum_{k=1}^{N-1} \left(x_k^T Q x_k + u_k^T R_{\varsigma_k} u_k \right) + x_N^T Q_N x_N, \quad (16)$$

where $R_0 = 1$, $R_1 = R_2 = 0.001$, $R_3 = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}$, $Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ and $Q_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The terminal set is $\mathcal{X}_f = \{x | x_N^T Q_N x_N \leq 0.2\}$. Satisfaction of Assumption 1 part 1) and 2) is immediately clear, and satisfaction of Assumption 1 part 3) and 4) and Assumption 2 can be shown using the S procedure.

4.3 Results

The four controllers performed as expected in the conducted experiments. In Figure 2 the realized cost, $x_t^T Q x_t + u_t^T R_{\sigma_t} u_t$, is shown accumulated over time. As expected the Optimal controller is over time the cheapest controller, while the Safe controller is the most expensive. Furthermore, the Optimal controller only accumulates a

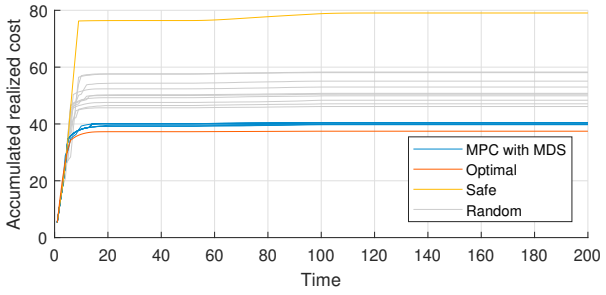

Fig. 2. The accumulated cost of the applied inputs and realized states under the different controllers.

little more cost when the system is disturbed between $t = 50$ and $t = 100$ while the Safe controller is costly. MPC with MDS performs well compared to both Safe and Random controllers. MPC with MDS performs significantly better than the Safe controller when the system is disturbed. The safe input was chosen by MPC with MDS mainly when the system was close to the reference value. During the disturbance ($50 \leq t \leq 98$), a better solution was found at 94,4% of the timesteps.

In the experiment, the maximum computation time for one iteration of the Optimal controller was 0.6032 s. MPC with MDS was implemented in serial, with a maximum computation time of 1.6575 s. However, the strength of MPC with MDS is its parallelizability. Computing the parts of the algorithm that must be serial took maximum 0.0189 s and the longest time it took to solve (10)-(14) was 0.3669 s. Computing Algorithm 2 in serial is time consuming due to the large amount of data. Updating the estimated fitness and uncertainty value took 0.4389 s in serial. The computation for each switching signal sequence is independent, and therefore easily parallelizable. It is expected that a parallel implementation of MPC with MDS reaches computation times comparable to that of the Optimal controller on this small system.

5. CONCLUSIONS AND FUTURE WORK

We have shown how ideas from Bayesian optimization can be used to parallelize the computations performed by MPC on switched linear systems. The presented method, MPC with Memory-based Discrete search (MDS) guarantees recursive feasibility and stability of the system. It finds 'good enough' inputs fast. The solution is not optimal, but the performance improvements compared to implementing a known safe input makes it an interesting method for many applications. It is expected that stability of MPC with MDS can be established for a broader class of systems, which is a promising future research.

In MPC with MDS, the computational complexity is decreased by reducing the mixed-integer MPC problem to a prespecified number of convex optimization problems. The results indicate that this decreases computation times, if the algorithm is implemented in parallel. How many parallel agents and how large a set of potential sequences to choose will be highly problem dependent and is another interesting line of future research. It is expected that, with sufficient parallelization, the computation time will increase less than that of a mixed-integer MPC when the system size increases.

REFERENCES

Axehill, D., Besselmann, T., Raimondo, D.M., and Morari, M. (2014). A parametric branch and bound approach to suboptimal explicit hybrid MPC. *Automatica*, 50(1), 240 – 246.

Gau, S., Leifeld, T., and Zhang, P. (2017). A novel and fast mpc based control strategy for switched linear systems including soft switching cost. In *Proc. of Conference on Decision and Control*, 6513–6518.

Jones, D.R., Schonlau, M., and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492.

Lazar, M. and Heemels, W. (2009). Predictive control of hybrid systems: Input-to-state stability results for sub-optimal solutions. *Automatica*, 45(1), 180 – 185.

Lin, H. and Antsaklis, P.J. (2009). Stability and stabilizability of switched linear systems: A survey of recent results. *Transactions on Automatic Control*, 54(2), 308–322.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *Proc. of International Symposium on Computer Aided Control System Design*.

Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789 – 814.

Mendes, P.R., Maestre, J.M., Bordons, C., and Normey-Rico, J.E. (2017). A practical approach for hybrid distributed mpc. *Journal of Process Control*, 55, 30 – 41.

Morari, M. and Bari, M. (2006). Recent developments in the control of constrained hybrid systems. *Computers & Chemical Engineering*, 30(10), 1619 – 1631.

Naik, V.V. and Bemporad, A. (2017). Embedded mixed-integer quadratic optimization using accelerated dual gradient projection. *IFAC-PapersOnLine*, 50(1), 10723 – 10728. 20th IFAC World Congress.

Pannocchia, G., Rawlings, J.B., and Wright, S.J. (2011). Conditions under which suboptimal nonlinear mpc is inherently robust. *Systems & Control Letters*, 60(9), 747 – 755.

Prada, C., Grossmann, I., Sarabia, D., and Cristea, S. (2009). A strategy for predictive control of a mixed continuous batch process. *Journal of Process Control*, 19(1), 123 – 137.

Sager, S., Bock, H.G., and Diehl, M. (2012). The integer approximation error in mixed-integer optimal control. *Mathematical Programming*, 133(1), 1–23.

Scokaert, P.O.M., Mayne, D.Q., and Rawlings, J.B. (1999). Suboptimal model predictive control (feasibility implies stability). *Transactions on Automatic Control*, 44(3), 648–654.

Snoek, J., Larochelle, H., and Adams, R.P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, 2951–2959.

Zhang, L., Zhuang, S., and Braatz, R.D. (2016). Switched model predictive control of switched linear systems: Feasibility, stability and robustness. *Automatica*, 67, 8 – 21.

Zhu, F. and Antsaklis, P.J. (2015). Optimal control of hybrid switched systems: A brief survey. *Discrete Event Dynamic Systems*, 25(3), 345–364.