Partial Enumeration MPC: Robust Stability Results and Application to an Unstable CSTR

Gabriele Pannocchia∗ James B. Rawlings ∗∗ Stephen J. Wright ∗∗∗

∗ Dept. of Chem. Engineering (DICCISM), Univ. of Pisa, Pisa, Italy (e-mail: g.pannocchia@diccism.unipi.it).
∗∗ Chemical and Biological Engineering Dept., Univ. of Wisconsin, Madison, WI, USA (rawlings@engr.wisc.edu)
∗∗∗ Computer Science Dept., Univ. of Wisconsin, Madison, WI, USA (e-mail: swright@cs.wisc.edu)

Abstract: We revise in this paper the Partial Enumeration (PE) method for the fast computation of a suboptimal solution to linear MPC problems. We derive novel robust exponential stability results for difference inclusions to show that the existence of a continuous Lyapunov function implies Strong Robust Exponential Stability (SRES), i.e. for any sufficiently small perturbation. Given the fact that the suboptimal PE-based control law is non-unique and discontinuous, i.e. a set-valued map, we treat the closed-loop system, appropriately augmented, as a difference inclusion. Such approach allows us to show SRES of the closed-loop system under PE-based MPC. Application to a simulated open-loop unstable CSTR is presented to show the performance and timing results of PE-based MPC, as well as to highlight its robustness to process/model mismatch, disturbances and measurement noise.

Keywords: Partial Enumeration MPC, Explicit MPC, Robust stability, Lyapunov functions

Notation. Given a vector $x \in \mathbb{R}^n$, $|x|$ denotes the 2-norm; given a positive scalar $r$, we define $B_r = \{x \in \mathbb{R}^n, |x| \leq r \}$; given a sequence of vectors $x_k = \{x(j)\}_{j=0}^{k-1}$, we define $|x_k| = \sup_j |x(j)|$; if $A$ is a subset of $\mathbb{R}^n$ and $b \in \mathbb{R}^n$, we denote the set $A + b = \{c + b | a \in A \}$.

1. INTRODUCTION

A significant amount of research activity of the last decade (or so) in the field of Model Predictive Control (MPC) has been devoted to the implementation of efficient methods for solving the associated constrained optimal control problem, which for linear systems subject to linear constraints and quadratic performance function can be casted as a Quadratic Program (QP). Several methods rely exclusively on efficient on-line calculations (Rao et al., 1998; Ferreau et al., 2008; Diehl et al., 2008), whereas so-called Explicit MPC methods (Alessio and Bemporad, 2008; Bemporad et al., 2002) move the most expensive calculations offline and the online computations are limited to a table lookup involving simple matrix/vector multiplications and inequality checks. However, due to the exponential explosion of the required number of table entries with respect to the problem size (number of inputs, states and constraints), Explicit MPC is limited to small systems. A method that can be considered in the middle field between Explicit MPC and online optimization is Partial Enumeration (Pannocchia et al., 2007, 2009), in which a table (with entries equivalent to those of Explicit MPC) of fixed size is scanned online to find the optimal control input. If none of the entries is optimal, a quick suboptimal input is computed, but the table is updated to include the new optimal entry for future decision times (while the least recently optimal entry is discarded). In this way, as the time goes on, the table adapts to the new operating conditions and contains only the entries that are currently more likely to be optimal.

The objective of this paper is to revise the Partial Enumeration (PE) approach and to make appropriate modifications with the goal of showing its robust stability properties. To this aim we develop novel tools for robust stability of suboptimal MPC, and this represents the second (perhaps the most relevant) contribution of this paper. The rest of this paper is organized as follows. In Section 2 we revise the PE-MPC method, and in Section 3 we derive from scratch novel robust stability tools and apply such results to PE-MPC. A simulated application to the control of an unstable CSTR is presented in Section 4, conclusions are drawn in Section 5.

2. PARTIAL ENUMERATION MPC

2.1 Control problem and main assumptions

We consider LTI systems with input constraints:

$$x' = Ax + Bu, \quad u \in U,$$

in which $u \in \mathbb{R}^m$ is the input, $x \in \mathbb{R}^n$ and $x^+ \in \mathbb{R}^n$ are the state and the successor state, respectively. $U = \{u | Du \leq b_u \}$ is compact and contains the origin in its interior (i.e. $b_u > 0$). We consider the problem of steering the state to a given target $\bar{x}$ that satisfies $\bar{x} = A\bar{x} + B\bar{u}$, with $\bar{u} \in U$. To this aim, we define deviation state and input: $\tilde{x} = x - \bar{x}, \tilde{u} = u - \bar{u}$, and consequently the deviation input admissible space is $\tilde{U} = \{\tilde{u} | D\tilde{u} \leq b_{\tilde{u}} - D\bar{u} \}$. In the sake of notation simplicity, we omit the dependance of $\tilde{U}$ (and its derived sets) on $\bar{u}$. Given a deviation input sequence vector $\tilde{u} = [\tilde{u}(0)', \tilde{u}(1)', \ldots, \tilde{u}(N-1)']'$, we define the following cost function:
\[ V_N(\tilde{x}, \tilde{u}) = \frac{1}{2} \sum_{k=0}^{N-1} \tilde{x}(k)' Q \tilde{x}(k) + \tilde{u}(k)' R \tilde{u}(k) + \frac{1}{2} \tilde{x}(N)' P \tilde{x}(N), \text{ s.t. } \tilde{x}+ = A \tilde{x} + B \tilde{u}, \tilde{x}(0) = \tilde{x}, \ldots \text{ parameters } (\bar{u}, \tilde{x}); \text{ candidate sequence } \tilde{u}+, \text{ its cost } V^+_{\text{NN}} \text{ if feasible (otherwise } V^+_{\text{NN}} = \infty); \text{ maximum table size } M_{\text{max}}. \]

Assumption 1. The pair \((A, B)\) is stabilizable, \((Q, R)\) are positive definite, \(P = S' S\) with \(P\) solution to the Lyapunov equation \(\Pi = A' \Pi A + S' Q S\), in which \((A, \Pi)\) come from the real Schur decomposition: \(A = [s, s, \ldots, A_n', A_n' \mid s_{n+1}']\), and \(A_n\) contains all stable eigenvalues of \(A\).

Remark 2. The constraint \(S'_{\text{NN}} \tilde{x}(N) = 0\) zeroes the unstable modes at time \(N\). Thus, \(V_N(\cdot)\) represents an infinite-horizon cost, under the control \(\tilde{u}(k) = 0\) (i.e. \(u(k) = \bar{u}\)) for all \(k \geq N\).

We define \(\tilde{X}\) as the set of \(\tilde{x}(0)\) for which problem (2) has a solution, i.e., \(\tilde{X} = \{\tilde{x} | \exists \tilde{u} : D_{\tilde{u}}(k) \leq b_u - D_u \text{ for all } k = 0, 1, \ldots, N-1 \text{ and } S'_{\text{NN}}(A_{\text{NN}}^i \tilde{x} + A_{\text{NN}}^{-1} B_{\tilde{u}}(0) + \cdots + B_{\tilde{u}}(N-1)) = 0\}\), and we make the following assumption on the input set and target.

Assumption 3. The set \(\tilde{U}\) is compact, and \(\tilde{u} \in \text{int } \tilde{U}\).

Remark 4. \(S_u\) is vacuous for stable systems, thus for any \(\tilde{u} \in \text{int } \tilde{U}\) it follows that \(\tilde{X} = \mathbb{R}^n\), and hence \(\tilde{X}\) contains the origin in its interior. For unstable systems \(S_u\) is a full rank matrix, and for any \(\tilde{u} \in \text{int}(\tilde{U})\) it is straightforward to show that \(\tilde{X}\) is nonempty and contains the origin in its interior.

2.2 Partial Enumeration: main definitions

As shown in (Pannocchia et al., 2010), problem (2) can be written as the following convex parametric QP:

\[
\begin{align*}
\min_{\tilde{u}} & \quad V_N(\tilde{x}, \tilde{u}) = \frac{1}{2} \tilde{x} + \tilde{u} - D_u - \tilde{u} + \tilde{u} + F \tilde{x} = 0. \\
\text{s.t.} & \quad \tilde{D}_{\tilde{u}} + C_{\tilde{u}} \leq d, \quad \tilde{E}_{\tilde{u}} + F \tilde{x} = 0. \quad (3a) \quad (3b)
\end{align*}
\]

Given the optimal point of this problem, \(\tilde{u}^0\), we denote with \((D_u, C_u, d_u)\) the stacked rows of \((D, C, d)\) such that \(D_{\tilde{u}}^0 + C_{\tilde{u}} \leq d_u\) (i.e. the active constraints). We also denote with \((D_u, C_u, d_u)\) the stacked complementary rows, i.e. such that \(D_{\tilde{u}}^0 + C_{\tilde{u}} < d_u\) (i.e. the inactive constraints). Since \(\tilde{u}^0\) is optimal for (3), the following first-order optimality KKT conditions hold:

\[
\begin{align*}
\tilde{H}_{\tilde{u}}^0 + G \tilde{x} + D_{\tilde{u}}^0 \lambda^0_{\tilde{u}} + E' \mu^0 = 0, \qquad (4a) \\
D_{\tilde{u}}^0 + C_{\tilde{u}} \leq d_u, \quad (4b) \\
E \lambda^0_{\tilde{u}} + F \tilde{x} = 0, \quad (4c) \\
\mu^0 \geq 0, \quad (4d) \\
D_{\tilde{u}}^0 + C_{\tilde{u}} \leq d_u. \quad (4e)
\end{align*}
\]

In (Pannocchia et al., 2010), we derive the following equivalent set of conditions for \(\tilde{u}^0\) satisfying (4):

\[
\begin{align*}
\tilde{u}^0 = \Gamma_u (d_u - C_u \bar{u}) + \Gamma_x \bar{x}, \quad \text{where} \\
\left[ \begin{array}{c}
\lambda^0_{\tilde{u}} \\
\mu^0
\end{array} \right] \leq \left[ \begin{array}{c}
\phi^0_{\tilde{u}} \\
\phi^0_{\infty}
\end{array} \right], \quad z \in [\ell, \bar{u}] \quad (5a)
\end{align*}
\]

Thus, we can express the optimal cost as follows:

\[
V_N(\tilde{x}, \bar{u}) = \frac{1}{2} \tilde{z}' \tilde{V}_z \tilde{z} + v_1 \tilde{z} + v_0. \quad (6)
\]

1 The main robust stability results presented in this paper can be extended to case in which \(\bar{u}\) lies on the boundary of \(\tilde{U}\), but several technical issues arise by doing so. For this reason such case is not treated in this paper.

Explicit MPC (Bemporad et al., 2002) partitions (offline) the space of \(z\) in a number of regions, each defined by the tuple:

\[
(P_f, P_d, \Gamma_u, \Gamma_x, \psi_f, \psi_d, V_2, v_1, v_0).
\]

The on-line evaluation consists in finding the region for which (5b) holds, and then computing \(\tilde{u}^0\) from (5a) and the optimal objective value from (6). Several enhancements can be made to reduce the storage requirements and also the online computations (Alessio and Bemporad, 2008). Still, Explicit MPC can be effectively implemented for small dimensional systems, as the number of regions grows exponentially with the problem size. On the other hand, in Partial Enumeration, PE (Pannocchia et al., 2007) we store the tuples (7) for a fixed number of active sets that were optimal at the most recent decision time points. Online, we scan the table to check if, for given parameters \((\bar{u}, \bar{x})\), the optimality conditions (5b) are satisfied, and in such case we compute the optimal solution from (5a).

However, given the fact that not all possible optimal active sets are stored, it is possible that no table entry is optimal. In such case we compute a suboptimal solution for closed-loop control. Nonetheless, a QP solver is called afterwards to compute the optimal solution \(\tilde{u}^0\), and thus derive the optimal missing tuple (7). Whenever, this table entry becomes available, it is inserted into the table. If after this insertion, the table would exceed its maximum size (user defined), we delete the entry that was optimal least recently. In this way, the table size is fixed and hence the table lookup process is fast, but the table entries are updated to keep track of new operating conditions for the systems. In (Alessio and Bemporad, 2008) a table with fixed number of entries is also proposed for fast evaluation, but differently from PE the table is not updated during online operation.

2.3 Partial Enumeration algorithms

In order to compute quickly a suboptimal input sequence when the table does not include the optimal active set for the current parameters \((\bar{u}, \bar{x})\), several options can be considered. To this end, a procedure based on violations of optimality conditions in (5) is developed in (Alessio and Bemporad, 2008), and closed-loop nominal stability is checked a posteriori. In (Pannocchia et al., 2007) we used the previous shifted optimal sequence in nominal conditions, thus guaranteeing nominal stability, or the solution to a short-horizon MPC problem in the presence of disturbances. Here, instead, we propose a slightly different procedure that allows us to prove robust exponential stability of the closed-loop under PE-MPC. The procedure requires two points, the first one of which needs to be feasible and its computation is discussed later in Algorithm 1. The second point, instead, is the minimizer of (3a) subject to the equality constraint (if present). More specifically, we define \(\tilde{u}^+\) as the solution to:

\[
\min_{\tilde{u}} V_N(\tilde{x}, \tilde{u}) \text{ s.t. } \tilde{E}_{\tilde{u}} + F \tilde{x} = 0. \quad (8)
\]

As discussed in (Pannocchia et al., 2010), we have that

\[
\tilde{u}^+ = \Gamma_u \tilde{z}, \quad (9)
\]

where the matrix \(\Gamma_u\) can be computed offline. Next, we denote by \(\hat{u}_{\infty} = [u^*(1) - \bar{u}, \ldots, u^*(N-1) - \bar{u}]^T\) the previous shifted optimal sequence vector, where the inputs \(u^*(1), \ldots, u^*(N-1)\) were computed at the previous decision time, while \(\hat{u}\) is the current input target. We now present the PE algorithm.

Algorithm 1. (General purpose PE). \textbf{Require}: Table with \(M\) entries, each a tuple of the form (7); current parameters \((\bar{u}, \bar{x})\); candidate sequence \(\hat{u}_{\infty}\), its cost \(V_{\infty}^C\) if feasible (otherwise \(V_{\infty}^C = \infty\)); maximum table size \(M_{\text{max}}\).
1: [%Initialize] Set opt\_found=false.
2: while (j ≤ M & opt\_found=false) do
3: Extract the j–th tuple from the table.
4: if Ψ≤E ≥ φθ then [%Entry is feasible]  
5: if Ψ≥E ≤ φθ then [%Entry is optimal]  
6: Compute optimal solution  from (5a). Put tuple j in first position of the table. Set opt\_found=true.
7: else [%Entry is infeasible]  
8: Compute cost  from (6).
9: if  then  
10: Set  = , , with  vector of ones. Redefine  = , (1−) + 0.  
11: end if
12: end if
13: end if
14: end while
15: if opt\_found=false then [%No optimal entry found]  
16: if  then [%No feasible solution]  
17: Solve the LP:  
18: end if
19: Evaluate  from (9), and compute the largest t ∈ [0,1] such that  
20: [%Table update, performed after returning  θ] Solve the QP (3), and find the optimal tuple (7). If  = , delete the entry that was optimal least recently (hence  +  − 1). Insert the new entry in first position of the table, set  = + 1.
21: end if
22: return (Sub)optimal sequence  , updated table.

Remark 5. The “feasibility recovery” step (Line 17) is required only if the system is open-loop unstable and either the target has changed from the previous decision time or a disturbance occurred. In the nominal case without target change, such step is not performed because  is feasible. Line 17 is the only “expensive” computation in Algorithm 1 and is justified by closed-loop stability reasons of an open-loop unstable system. Also notice that Line 19 computes the largest feasible step from  to  ; if  is feasible it follows that  = 1.

If the input constraints are: , i.e. the constraint matrix/vector are given by  = , , then we propose a tailored enhanced strategy.

Algorithm 2. (Enhanced PE for box constraints). Same as Algorithm 1, with Lines 15–19 replaced by the following.

1: Set  , empty matrix/vector. Set feas\_found=false.
2: while feas\_found=false do
3: Solve the following (equality-constrained) QP:  
4: if  then [%Feasible solution found]  
5: Define  = . Set feas\_found=true.
6: else if  ≤  then  
7: Set  = .  
8: else  
9: Set  = .  
10: end if
11: end if  
12: Define  as the rows of  for the violated inequalities plus the rows of the previous  with nonnegative multipliers.
13: end if
14: end while

Remark 6. In Algorithm 2, when  violates any inequality such constraints are regarded as equalities, and  is recomputed. When a given constraint (say an upper bound) is included in  , there is no need to check feasibility with respect to the parallel (say lower bound) constraint. Also notice that (10) reduces to solving a square linear system.

3. ROBUST STABILITY RESULTS

In (Grimm et al., 2004), Teel and coworkers showed that under standard assumptions, the origin of a linear closed-loop system  =  +  with  being a nominally stabilizing MPC-generated control law, is robustly asymptotically stable. This result relies on continuity of  on , which however holds only when the optimal solution to the MPC problem is attained. Unfortunately, the suboptimal MPC law is not continuous, even for linear systems, and furthermore it is not a unique function of the state  as it also depends on the initial guess input sequence. These facts prevent us from establishing freely even nominal stability. This point was discussed in (Scokaert et al., 1999) in the context of nonlinear MPC to show that suboptimal nonlinear MPC, under appropriate restrictions, is nominally asymptotically stabilizing. We note that the suboptimal input computed by PE also depends on the entries contained in the working table and hence the outcome for the same state and initial guess input sequence may be different with different working tables. Thus, in this paper we treat the suboptimal MPC law as a set-valued map, and we derive from scratch novel results for robust exponential stability of difference inclusions (Kellett and Teel, 2004) and show that such results apply to PE-MPC.

3.1 General stability results for difference inclusions

Let  be a set-valued map from  to subsets of  with  being the equilibrium point, i.e.  = ; let  be a solution at time  of the difference inclusion  =  with  being the solution of a starting condition  =  ∈ . Rawlings and Mayne (2009, pp. 196–203) provide an introduction to MPC with difference inclusions. We also consider a perturbed difference inclusion  =  + , and we denote with  a solution to the perturbed difference inclusion at time  with initial condition  =  ∈ for given state and additive disturbance sequence  = , hence  =  + .

Definition 7. (Exponential Lyapunov function). A function  :  →  is an exponential Lyapunov function in the set  for the difference inclusion  =  if there exist positive scalars , such that  ∈  implies that:

\[ a_1|\xi|^p \leq V(\xi) \leq a_2|\xi|^q, \quad \max_{\xi \in \mathbb{E}(\xi)} V(\xi) \leq V(\xi) - a_3|\xi|^p. \]

Definition 8. (Exponential Stability). The origin of the difference inclusion  =  is said to be exponentially stable (ES) on  if there exist positive scalars  and  such that for any  ∈ all solutions  satisfy:

\[ \phi(\xi, k) \in \Xi, \quad |\phi(\xi, k)| \leq b_4|\xi| \quad \text{for all } k \in \mathbb{Z}_0. \]

We have the following result.

Lemma 9. If the set  is positively invariant for difference inclusion  =  and there exists an exponential Lyapunov function  in , then the origin is ES on .

Proof. From the definition of  , we have that  ∈  implies:

\[ \max_{\xi \in \mathbb{E}(\xi)} V(\xi) \leq V(\xi) - a_3|\xi|^p \leq V(\xi) - a_3a_2V(\xi) \leq \gamma V(\xi), \]

with  = 1 − 2/(2a2). Notice that  ≥ , hence 0 <  < 1. Since  =  ∈ for all  , we can write: \[ |\phi(\xi, k)|^q \leq \frac{V(\phi(\xi, k))}{a_1} \leq \]
From this, we obtain: |φ(ξ, k)| ≤ bλk|ξ| in which λ = γ1/a and b = (a2/a1)1/a, and we observe that 0 < λ < 1. □

**Definition 10. (Robust Exponential Stability).** The origin of the difference inclusion ε* = F(ξ) is said to be robustly exponentially stable (RES) on Ξ, ξ ∈ Ξ, with respect to state and additive disturbances if there exist positive scalars b and λ, λ < 1, and for each ε > 0 there exists δ > 0 such that for all ξ ∈ Ξ and all disturbance sequences e, p satisfying:

\[ 0 <\max(||e||, ||p||) ≤ \delta, \quad \phi(\xi, k) \in \Xi \text{ for all } k, \]

the perturbed solutions φ̃p(ξ, ·) satisfy |φ̃p(ξ, k)| ≤ bλk|ξ| + ε.

**Lemma 11.** If there exists a continuous exponential Lyapunov function V on Ξ, then the origin of the difference inclusion ε* = F(ξ) is RES on Ξ w.r.t. state and additive disturbances.

**Proof.** From the proof of Lemma 9, we have that for all ξ ∈ Ξ \ 0 the following strict inequality holds: maxε∈F(ξ) V′(ε) < γV(ξ) for some γ < 1. We now require the following Lemma proved in (Pannocchia et al., 2010).

**Lemma 12.** For every μ > 0, there exists δ > 0 such that, for all (e, p) ∈ Ξ × δξ × δΞ that satisfy ε ∈ Ξ and F(ε) + p ∈ Ξ, the following conditions hold: V′(ε) ≤ max(γV(ε), μ) for all ε ∈ F(ε) + p.

Now, assume that φ̃p(ξ, k) ∈ Ξ for all k ∈ Ξ. Then, by induction we can write: a1|φ̃p(ξ, k)|2 ≤ V(φ̃p(ξ, k)) ≤ maxγV(ε), μ ≤ max|φ̃p(ξ, k)| ≤ max{a2|ξ|2, (μa1)|ε|2} in which λ = γ1/a and b = (a2/a1)1/a. Finally, we define ε = (μa1)|ε|2 and write: |φ̃p(ξ, k)| ≤ max{a2|ξ|2, (μa1)|ε|2 ≤ bλk|ξ| + ε, which completes the proof by noticing, as in the proof of Lemma 9, that λ < 1. □

**Remark 13.** The definition of RES and also that of Robust Asymptotic Stability (RAS) in (Grimm et al., 2004) assume that there exist “nice” and small perturbation sequences e, p such that the perturbed solutions φ̃p(ξ, ·) remain in Ξ at all times. The next definition only assumes that the perturbations are sufficiently small, while feasibility of φ̃p(ξ, ·) is implied.

**Definition 14. (Strong Robust Exponential Stability).** The origin of the difference inclusion ε* = F(ξ) is said to be strongly robustly exponentially stable (SRES) on Ξ, ξ ∈ Ξ, with respect to state and additive disturbances if there exist a compact set C ⊆ Ξ and positive scalars b and λ, λ < 1, and for each ε > 0 there exists δ > 0 such that for all e, p satisfying:

\[ \max(||e||, ||p||) ≤ \delta, \]

for all ξ ∈ C the perturbed solutions φ̃p(ξ, ·) satisfy:

\[ \phi(\xi, k) \in \Xi \text{ for all } k, \quad |\phi(\xi, k)| ≤ bλk|ξ| + ε. \]

**Theorem 15. (Strong Robust Exponential Stability).** If Ξ ∈ Ξ and there exists a continuous exponential Lyapunov function V on Ξ, then the origin of the difference inclusion ε* = F(ξ) is SRES on Ξ with respect to state and additive disturbances.

**Proof.** Let c be a positive scalar such that C = {ξ ∈ Rn \ V(ξ) ≤ c \ Ξ, i.e. C is a sublevel set contained in Ξ. Notice that Ξ is compact and that 0 ∈ int C. From the proof of Lemma 9 and from Lemma 12 we have that for each μ > 0, there exists a δ1 > 0 such that the following conditions hold:

\[ \max_{ε∈F(ε)+p} V′(ε) ≤ \max(γV(ε), μ) < γV(\xi) + μ \]

for all (ξ, e, p) ∈ Ξ × δ1B × δ1B that satisfy e ∈ Ξ and F(ε) + p ∈ Ξ, in which 0 < c < 1. We also have that for each μ > 0, the condition maxε∈F(ε)+,p V′(ε) < γV(ε) + μ holds for all (ξ, e, p) ∈ Ξ × δ1B × δ1B that also satisfy V(ξ) > r’ = γ1/a.

Define the sublevel set R = {ξ : V(ξ) ≤ r’} and choose μ and ρ sufficiently small that R ⊆ C (hence R ⊆ Ξ). We now observe that for all e, p satisfying max(||e||, ||p||) ≤ δ, the following conditions hold: (i) if ξ ∈ R, it follows that maxε∈F(ε)+,p V′(ε) ≤ γr’ + μ ≤ γr’ + ρ, and hence the solutions φ̃p(ξ, k) remain in R for all k ∈ Ξ. (ii) if ξ ∈ C \ R, the solutions φ̃p(ξ, k) remain in C for all k ∈ Ξ and enter R in finite time. This proves that there exists a compact set C ⊆ Ξ such that if ξ ∈ C the solutions φ̃p(ξ, k) remain in C at all times. We can apply Lemma 11 to show that there exist positive scalars b and λ, λ < 1, and for each ε > 0 there exists a δ, 0 < δ ≤ δ1 such φ̃p(ξ, k) ≤ bλk|ξ| + ε for all e, p satisfying max(||e||, ||p||) ≤ δ.

**3.2 Nominal and robust stability of Partial Enumeration MPC**

**Lemma 16.** The set Ξ is positively invariant for the closed-loop system x̄ = Ax̄ + B(ū) + ˜u.

**Proof.** Let ˜u = (u(0), u(1), . . . , u(N − 1)) be the solution computed by Algorithm 1 for the initial state x̄(0) = x̄, and assume that x̄ ∈ Ξ. Then, for x̄ = Ax̄ + B(ū) we consider the sequence ˜u = (u(0), u(1), . . . , u(N − 1), 0), and observe that it is feasible w.r.t. to D̄u ≤ b̄u − D̄u and such that S′(Ax̄ + ÃN−1B̄u + · · · + B̄u(N − 1) = 0. Thus, x̄ ∈ Ξ.

**Lemma 17.** There exists a constant c > 0 such that the (sub)optimal solution ˜u computed by Algorithm 1 for the initial state x̄(0) = x̄ satisfies |u| ≤ c|x| for all x̄ ∈ Ξ.

**Proof.** We first consider any x̄ ∈ Ξ, x̄ ∈ Ξ and show that the result holds provided that r > 0 is sufficiently small. If the optimal solution is found in the table, i.e. from Line 6 and when x̄ is sufficiently small, the optimal solution ˜u coincides with the “unconstrained” minimizer ˜u = ˜ūx because such point gives the lowest cost and it is feasible w.r.t. to the constraint: D̄u + C̄u ≤ d̄u because 0 < d − C̄u. If instead a suboptimal solution is computed from Line 19, since ˜u is feasible for x̄ sufficiently small we have that t = 1, i.e. the computed solution is again the “unconstrained” minimizer. Therefore, there exist positive constants c’ and r such that |u| ≤ c’|x| for all x̄ ∈ Ξ.

To complete the proof, we take account of the remaining case in which |x| ≥ r. We define η = maxω∈Ω|ū(ω)| (which is finite because U is compact and so are ˜ū and Ūω) and set c = max(c’, η/r). We then have that ˜ū computed by Algorithm 1 satisfies the required inequality |u| ≤ c|x| because: (i) for |x| < r we have |u| ≤ c’|x| ≤ c|x| as discussed above; (ii) for |x| ≥ r, we have |u| ≤ η(|x|)/r ≤ c|x|.

**Theorem 18.** (ES under PE-MPC). The origin of the closed-loop system x̄ = Ax̄ + B̄u(0), is ES on Ξ.

**Proof.** We denote by ˜ū the suboptimal solution computed for the successor state x̄, and we observe that: x̄ = Ax̄ + B̄I 0|ū and ˜ū ∈ G(x̄, ˜ū) in which G(·) is a set-valued map, because ˜ū depends on: the state x̄ = Ax̄ + B̄I 0|ū, the initial guess obtained by shifting ˜ū but also on the current working table. The following conditions hold. i) V̄(x̄, ˜ū) ≥ a1[(x̄, ˜ū)]2.
for some \( a_1 > 0 \) because \( V_N() \) is quadratic and strictly convex in its arguments, as discussed in (Pannocchia et al., 2010). Similarly, there exists some \( a_2 > 0 \) such that \( V_\delta(\tilde{x}, \tilde{u}) \leq a_2|\tilde{x}|^2 \). ii) There exists a \( c > 0 \) such that \( |u| \leq c|x| \) for all \( \tilde{x} \in \mathbb{R} \), from Lemma 17. iii) \( V_N(\tilde{x}, \tilde{u}) - V_N(\tilde{x}, \tilde{u}) \leq \frac{1}{2}(\tilde{x}'Q\tilde{x} + u(0)R(0)u(0)) \leq -a_3\tilde{y}(\tilde{x}, u(0)) - a_3|\tilde{x}|^2 \leq -a_3(\tilde{x}, u(0)) \text{ for } a_3 = \frac{1}{2}a_3 \text{ min}[1, 1/|\tilde{c}|^2] \) for some \( a_3 > 0 \), where in the last inequality we have used that \( |\tilde{x}| \geq \frac{1}{2}b|u| \). Thus, we have that \( V_N(\tilde{x}, u) \) is an exponential Lyapunov function for the extended closed-loop system, expressed as a difference inclusion: \((\tilde{x}', \tilde{u}^*) \in F(\tilde{x}, u)\).

We also have that \( \mathbb{R} \times \mathbb{R}^N \) is forward invariant for the extended closed-loop difference inclusion. Hence, there exist positive scalars \( \tilde{b} \) and \( \lambda, \lambda < 1 \), such that for all initial extended state \((\tilde{x}, u) \in \mathbb{R} \times \mathbb{R}^N \) the following condition holds for all \( k \in \mathbb{Z}_0^+ \):

\[
|\tilde{x}(k), u(k)| \leq \frac{\tilde{b}^k}{\lambda}(\tilde{x}, u(0)).
\]

If we denote \( \phi_0(\tilde{x}, k) = \tilde{x}(k) \), we can now write: \(|\phi_0(\tilde{x}, k) - \phi_0(\tilde{x}, 0)| \). We say that \( \tilde{x}(0) = \tilde{x}\) is a feasible solution. If we denote with \( \phi_{vp}(\tilde{x}, k) = \tilde{x}(k) \), there exist positive scalars \( \tilde{b} \) and \( \lambda, \lambda < 1 \), and for each \( c > 0 \), there exists a \( \delta > 0 \) such that for any \( (\tilde{x}, u(0)) \in \mathbb{C} \times \mathbb{R}^N \), we can write: \(|\phi_{vp}(\tilde{x}, k), u(k)| \). Notice that we can express the compact set appearing in the definition of SRES as \( \mathbb{C} \times \mathbb{R}^N \) because \( \mathbb{V} \) is compact and the PE algorithm always computes a feasible solution. For any \( \tilde{x} \in \mathbb{C} \), we can now write: \(|\phi_{vp}(\tilde{x}, k) - \phi_{vp}(\tilde{x}, 0)| \). We say that \( \tilde{x}(0) = \tilde{x}\) is a feasible solution.
Table 1. Performance and timing results: non-linear plant with disturbances.

<table>
<thead>
<tr>
<th>Controller</th>
<th>J</th>
<th>Aver. CPU time</th>
<th>Max. CPU time</th>
<th>Opt. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP-MPC</td>
<td>497.2</td>
<td>381 ms</td>
<td>27800 ms</td>
<td>–</td>
</tr>
<tr>
<td>PE1-MPC</td>
<td>500.4</td>
<td>4.2 ms</td>
<td>138 ms</td>
<td>0.907</td>
</tr>
<tr>
<td>PE10-MPC</td>
<td>500.4</td>
<td>2.6 ms</td>
<td>139 ms</td>
<td>0.960</td>
</tr>
<tr>
<td>PE25-MPC</td>
<td>500.4</td>
<td>2.0 ms</td>
<td>140 ms</td>
<td>0.977</td>
</tr>
<tr>
<td>PE50-MPC</td>
<td>500.4</td>
<td>1.9 ms</td>
<td>148 ms</td>
<td>0.980</td>
</tr>
<tr>
<td>PE200-MPC</td>
<td>500.4</td>
<td>1.8 ms</td>
<td>152 ms</td>
<td>0.983</td>
</tr>
</tbody>
</table>

Table 2. Performance and timing results: linear plant without disturbances.

<table>
<thead>
<tr>
<th>Controller</th>
<th>J</th>
<th>Aver. CPU time</th>
<th>Max. CPU time</th>
<th>Opt. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP-MPC</td>
<td>274.5</td>
<td>118 ms</td>
<td>22800 ms</td>
<td>–</td>
</tr>
<tr>
<td>PE1-MPC</td>
<td>273.4</td>
<td>1.5 ms</td>
<td>74.2 ms</td>
<td>0.979</td>
</tr>
<tr>
<td>PE10-MPC</td>
<td>273.4</td>
<td>1.5 ms</td>
<td>76.5 ms</td>
<td>0.980</td>
</tr>
<tr>
<td>PE25-MPC</td>
<td>273.4</td>
<td>1.5 ms</td>
<td>73.5 ms</td>
<td>0.981</td>
</tr>
<tr>
<td>PE50-MPC</td>
<td>273.4</td>
<td>1.2 ms</td>
<td>76.5 ms</td>
<td>0.991</td>
</tr>
<tr>
<td>PE200-MPC</td>
<td>273.4</td>
<td>1.2 ms</td>
<td>76.8 ms</td>
<td>0.991</td>
</tr>
</tbody>
</table>

0.005 (i.e., on average, one every 10 minutes), whereas random disturbances on $F_i$, $c_A$, and $T_i$ occur with a probability of 0.05 (i.e., on average, one every minute). To compare the performance of the different controllers we evaluate the closed-loop cost over the simulation period: $J = \frac{1}{N} \sum_{k=0}^{N-1} (y'(k) - y'(k))' (y(k) - y'(k)) + (u(k) - \bar{u}(k))^T R (u(k) - \bar{u}(k))$ in which $y(k) = C(x(k) + C_d(d(k)$ is the reachable output target at time $k$ and $\bar{u}(k)$ is the corresponding input target. Performance results are reported in Table 1, along with the average and maximum CPU time required to solve (2) (using Octave 3.2.3 on a 2.53 GHz MacBook Pro), and with the optimality rate. We also report in Table 2 the corresponding results that one would obtain in the nominal case, i.e., when the plant is exactly (12) and no disturbances and/or noise are present.

From Tables 1 and 2 we immediately observe that the difference in performance between using optimal (QP-based) and suboptimal (PE-based) is negligible. Furthermore all (optimal and suboptimal) controllers show a large degree of robustness (see Table 1). It is also interesting to observe that when the table size increases, as expected, the optimality rate increases. The differences in average and maximum CPU times between QP-based and PE-based MPC are particularly noticeable (roughly two orders of magnitude). The effect of the table size on timing is a bit more complicated to analyze. While a larger table slightly increases the maximum CPU time, it decreases the average CPU time. To understand these results it must be kept in mind that the table scanning time is only a fraction of the overall CPU time, as a relevant portion of it is spent to compute a suboptimal solution (as detailed in Algorithm 2) when the optimal solution is not in the table. Thus, since the use of a larger table implies a higher optimality rate, PE resorts less frequently to the suboptimal computation steps. Nonetheless, $J$ is essentially identical for all PE-based MPC, and this occurs because the sub-optimal steps described by Algorithm 2, which are executed when the current table does not contain the optimal tuple, often returns the optimal solution.

In this paper we revisited the Partial Enumeration (PE) method for solving more efficiently the constrained optimal control problem that arises in linear MPC, especially for large-scale systems (Pannocchia et al., 2007, 2009). We proved here that such suboptimal MPC algorithm is nominally exponentially stabilizing, and most importantly it is exponentially stabilizing in the presence of arbitrary (but sufficiently small) perturbations. Such novel results are based on considering the (non-unique and discontinuous) suboptimal control law $u = \kappa_0(x)$ as a set-valued map. Consequently, the closed-loop system is described by a difference inclusion, and we proved in this paper nominal and robust exponential stability for arbitrary perturbations under two main assumptions: the origin of the closed-loop system is in the interior of its region of attraction and a continuous Lyapunov function exists for the difference inclusion. Then, we showed that the closed-loop system (appropriately augmented) obtained using the proposed PE-MPC algorithm satisfies such assumptions.

We presented an application to the control of a nonlinear unstable CSTR in which the designed suboptimal (PE-based) controllers successfully faced the perturbations inherently generated by the nonlinear process vs. linear model mismatch, by process parameter disturbances and by measurement noise. Despite the presence of such unsettling disturbances closed-loop stability is maintained and the performance difference with respect to optimal (QP-based) MPC is negligible, while the CPU times are two orders of magnitude lower.

REFERENCES