A Dual-level Model Predictive Control Scheme for Multi-timescale Dynamical Systems

Xinglong Zhang, Wei Jiang, Zhizhong Li, and Xin Xu

Abstract—This paper presents a dual-level model predictive control (MPC) scheme for two-timescale dynamical systems subject to input and state constraints, with the scope to enforce closed-loop separable dynamics. A novel dual-level MPC (i.e., D-MPC) algorithm is initially presented. At the high level of the control structure, a stabilizing MPC regulator minimizes the deviation of the output and its setpoint at a slow time scale. A shrinking horizon MPC is designed at the low level to refine the deviation of the output and its setpoint at a fast time scale so as to generate satisfactory short-term transient of the output associated with the computed control actions in the basic time scale. To further improve the closed-loop control performance, an incremental D-MPC algorithm is also proposed, via introducing at the high level of the D-MPC an integral action, and an explicit design of the “fast” output reference. The proposed algorithms are not only suitable for systems characterized by different dynamics, but also capable of imposing separable closed-loop performance for dynamics that are non-separable and strongly coupled. The recursive feasibility and convergence properties of the D-MPC and incremental D-MPC closed-loop control systems are proven. The simulation results concerning the use of the proposed approaches for the control of a Boiler Turbine (BT) system, including the comparisons with a decentralized PID controller and a multirate MPC, are reported to show the effectiveness of the proposed algorithms in imposing closed-loop separable dynamics and the advantages in generating satisfactory control performance.

Index Terms—Model predictive control, dual-level, linear systems, separable dynamics, Boiler Turbine control.

I. INTRODUCTION

Many industrial processes are characterized by separable fast-slow dynamics, which can be called “multi-timescale dynamical systems”. In a multi-timescale dynamic system, for a given constant input signal, some of the output variables reach their steady-state values fast while the other ones may experience a longer transient time period, see for instance [11], [12], [13]. A widely-acceptable approach for the control of such systems consists in resorting to hierarchical control synthesis that possibly relies on singular perturbation theory (see the book [4]), where time-scale separation technique is adopted to define regulators at different control frequencies so as to guarantee the stability and performance of the dynamics associated with the each channel. As another example, there might be systems that their dynamics are not separable but must be controlled in the same way with a multi-rate control setting, see for instance the control of a Boiler Turbine (BT) system considered in [5]. In this case, usually, the crucial controlled variables of the considered system must be adjusted in a faster rate to meet the control performance requirement, while other outputs can be controlled more smoothly in a slower time scale.

Model predictive control (MPC) is an advanced process control technique, widely used in industrial processes, such as chemical plants and smart grids, see [6], [7], [8], [9], [10], in robotics, see [11], [12], in urban traffics, see [13], and in computing data center, see [14]. In MPC, the control problem is reformulated as an optimization one, that has to be solved on-line iteratively. This allows to explicitly consider the control, state and output constraints in the control problem. At any generic time instant $k$, an finite horizon optimization problem must be solved to compute the optimal control sequence. Only the first element is applied to the system, and the state and output variables are updated. The optimization problem is then repeated at the next instant $k + 1$. This is the so called “receding horizon” strategy. As an alternative, in the context of MPC, “shrinking horizon” strategy can also be used. The resultant algorithm is usually called “shrinking horizon MPC”, in which, the on-line optimization problem is still solved recursively, however the prediction horizon reduces with the time to go. This re-optimization nature paves the way for endpoint tracking objective and for disturbance/uncertainty effect reduction. see [15], [16].

In the framework of MPC, many solutions have been developed based on time-scale separation technique for systems characterized by open-loop separable dynamics. Among them, solutions on theoretical developments can be found in [17], [18], [19] for nonlinear singularly perturbed systems, in [20], [21], [22] for linear systems, and in [23] for input/output models, while the results focusing on the application aspect are reported in [24], [25], [26], [27]. However, most of the aforementioned works are tailored for systems with clearly different dynamics due to their dependencies on singular perturbation theory, so not suitable for the multirate control of systems with non-separable open-loop dynamics. Motivated by this, the scope of this work concerns designing dual-level algorithms based on MPC for linear multi-timescale dynamical systems subject to input and state constraints, such that the resulting closed-loop systems exhibit separable dynamic behaviors. A novel dual-level MPC (D-MPC) algorithm...
is initially presented. At the high level of the control structure, a slow time scale associated with $N$ period of the basic time scale is adopted to define a stabilizing MPC problem with respect to the sampled version of the original system so as to ensure convergence at the considered time scale. At the low level, a shrinking horizon MPC is designed in the basic time scale to refine the computed control actions in order to derive satisfactory short-term transient associated with the closed-loop fast dynamics and to verify endpoint state constraint computed from the high level. In doing so, the recursive feasibility and stability properties of the closed-loop system can be guaranteed. Moreover, an incremental form of D-MPC (i.e., incremental D-MPC) is proposed with an emphasis on the controller modification at the high level, including specifically an integral action on the “slow” outputs and, a prior explicit design of the reference trajectory of the “fast” inputs allowing to be furtherly refined in the fast control channel with the slow dynamics to be manipulated in the slow control algorithm described in [28] only permits the input associated with the considered system has no poles larger than 1. iii) The control law must be designed primarily (see the Section V). Examples that have poles on the unitary disk, a stable feedback system in [28] is assumed to be strictly stable. In this case, for the usage of impulse response representation, the concerned closed-loop recursive feasibility and stability properties of the closed-loop system can be guaranteed. Moreover, an incremental form of D-MPC (i.e., incremental D-MPC) is proposed with an emphasis on the controller modification at the high level, including specifically an integral action on the “slow” outputs and, a prior explicit design of the reference trajectory of the “fast” inputs allowing to be furtherly refined in the fast control channel with the slow dynamics to be manipulated in the slow control algorithm described in [28] only permits the input associated with the considered system has no poles larger than 1.

Whereas in this paper this restriction is slightly relaxed, i.e., however the control scheme described in this paper shows a significant improvement for the following reasons: i) The system to be controlled is described by a discrete-time linear system consisting of two interacting subsystems expressed as

$$
\Sigma_x: \begin{cases} 
x_s(h+1) = A_{ss}x_s(h) + A_{sf}x_f(h) + B_{ss}u_s(h) + B_{sf}u_f(h) 
\end{cases}
\quad (1a)
$$

$$
\Sigma_f: \begin{cases} 
x_f(h+1) = A_{fs}x_s(h) + A_{ff}x_f(h) + B_{fs}u_s(h) + B_{ff}u_f(h) 
\quad \text{(1b)}
\end{cases}
$$

where $u_s \in \mathbb{R}^{m_s}$, $x_s \in \mathbb{R}^{n_s}$, $y_s \in \mathbb{R}^{p_s}$, are the input, state, and output variables belonged to $\Sigma_x$, while $u_f \in \mathbb{R}^{m_f}$, $x_f \in \mathbb{R}^{n_f}$, $y_f \in \mathbb{R}^{p_f}$, are the ones associated with $\Sigma_f$, $h$ is a basic discrete-time scale index, the matrices $A_{ss}$, $B_{ss}$ (where $*$ is $sf$ or $fs$ in turn) represent the couplings between $\Sigma_x$ and $\Sigma_f$ through the state and input variables respectively.

Similar to [28], in this paper, models (1a) and (1b) are assumed to satisfy at least one of the following scenarios:

- $\Sigma_x$ is characterized by a slower dynamics in contrast to $\Sigma_f$ in the sense that some of the triples $(u_f, x_f, y_f)$ reach their final steady-state values fast while the other ones, i.e., $(u_s, x_s, y_s)$ may have begun their main dynamic motions, see the examples in [11, 12, 13];
- even if the dynamics of $\Sigma_x$ and $\Sigma_f$ might not be strictly separable, however they must be controlled in a multi-rate fashion, e.g., the triples $(u_f, x_f, y_f)$ must react promptly to respond to operation (reference) variations while the triples $(u_s, x_s, y_s)$ can be controlled in a smoother fashion, see for instance [13].

Combining (1a), (1b), the overall system is written as

$$
\Sigma: \begin{cases} 
x(h+1) = Ax(h) + Bu(h) 
y(h) = Cx(h)
\end{cases}
$$

where $u = (u_s, u_f) \in \mathbb{R}^{m}$, $m = m_s + m_f$, $x = (x_s, x_f) \in \mathbb{R}^{n}$, $n = n_s + n_f$, $y = (y_s, y_f) \in \mathbb{R}^{p}$, $p = p_s + p_f$. The diagonal blocks of the collective state transition matrix $A$ and input matrix $B$ are $A_{ss}$, $A_{sf}$ and $B_{ss}$, $B_{sf}$ respectively; whereas their non-diagonal blocks correspond to the coupling terms of the state and input variables between $\Sigma_x$ and $\Sigma_f$. The collective output
matrix is $C = \text{diag}(C_{cr}, \, C_{ff})$.

The following assumption is assumed to hold:

Assumption 1:

1. $A$ is stable, i.e., all the eigenvalues of $A$ are in the unitary disk $\mathcal{D} = \{z \in \mathbb{C}| |z| \leq 1\}$;
2. $m = p$, $m = p_s$, and the system (2) has no invariant zeros in 1, i.e., $\det(\Phi) \neq 0$ where
   \begin{equation}
   \Phi = \begin{bmatrix}
   I - A & -B \\
   C & 0 
   \end{bmatrix}.
   \end{equation}

The control objectives to be achieved are introduced here.

(i) **Output tracking**: for a given reference value $y_r = (y_{sr}, y_{fr})$, we aim to drive
   \begin{align}
   y_s(h) &\rightarrow y_{sr}, \quad (3a) \\
   y_f(h) &\rightarrow y_{fr}, \quad (3b)
   \end{align}

(ii) **Input and state constraints**: enforce the input and state constraints of the type
   \begin{align}
   u_s(h) &\in \mathcal{U}_s, \quad (4a) \\
   u_f(h) &\in \mathcal{U}_f, \quad (4b) \\
   x_s(h) &\in \mathcal{X}_s, \quad (4c) \\
   x_f(h) &\in \mathcal{X}_f \quad (4d)
   \end{align}

where $\mathcal{X}_s, \mathcal{X}_f, \mathcal{U}_s, \mathcal{U}_f$ are convex sets. Thanks to Assumption 1, from (2), it is possible to compute the steady-state input and state, i.e., $u_r = (u_{sr}, u_{fr}), x_r = (x_{sr}, x_{fr})$ such that $y_r = C x_r$ and $x_r = A x_r + B u_r$.

It is assumed that the sets $\mathcal{U} = \mathcal{U}_s \times \mathcal{U}_f$, $\mathcal{X} = \mathcal{X}_s \times \mathcal{X}_f$ contain $u_r, x_r$ in their interiors respectively.

In principle, a centralized MPC problem with respect to $\Sigma$ can be solved so as to achieve the above objectives. However, the resulting control performance might be hampered in the aforementioned scenarios due to the conflicting requirements of the sampling period and prediction horizon for $\Sigma_s$ and $\Sigma_f$ respectively. For instance, the control of the associated dynamics $\Sigma_f$ needs a higher input update frequency to ensure the short-term dynamic behaviors, while a larger prediction horizon might be expected for $\Sigma_s$ to guarantee the feasibility and stability of the adopted algorithm in the long term.

For this reason, a dual-level MPC (D-MPC) control scheme is initially proposed in this paper to fulfill the aforementioned control objectives. At the high level, a slow time scale $k$ associated with $N$ ($N \in \mathbb{N}$) period of the basic time scale $h$ is adopted to define a stabilizing MPC problem with respect to the sampled version of $\Sigma$ penalizing the deviation between the output and its setpoint. The computed values of the control actions at this level (i.e., $u^{[s]}(k), u^{[f]}(k)$) are held constant within the long sampling time interval $[kN, kN + N]$, i.e., $\bar{u}_s(h) = u^{[s]}(k), \bar{u}_f(h) = u^{[f]}(k)$ for all $h \in [kN, kN + N]$. At the low level, a shrinking horizon MPC is designed at the basic time scale to refine control actions with additional corrections (i.e., $\delta u_f(h), \delta u_s(h)$) in order to derive satisfactory short-term transient associated with the closed-loop fast dynamics and to verify endpoint state constraint computed from the high level. The overall control actions of the D-MPC regulator are described by

\begin{align}
   &u_s(h) = \bar{u}_s(h) + \delta u_s(h), \quad (5a) \\
   &u_f(h) = \bar{u}_f(h) + \delta u_f(h) \quad (5b)
\end{align}

Moreover, with the objective of further improving the control behavior of the fast and slow controlled variables, the incremental D-MPC algorithm is also proposed (see Section IV), in which the MPC at the high level is modified. To be specific, this version includes at the high level an integral action on the controlled variable $y$, and a prior explicit design of the output trajectory $y_f$ relying on an auxiliary optimization variable to be optimized at the slow time scale with the objective to enforce $y_f$ to steer to the reference value or its neighbor promptly. A brief diagram of the proposed approaches is displayed in Fig. 1.

**Fig. 1. A brief diagram of the proposed control scheme:** HMPC (LMPC) stands for the MPC at the higher (lower) level, iHMPC represents the incremental HMPC, while ZOH is the zero order holder.

### III. D-MPC ALGORITHM

In this section, the D-MPC algorithm consisting of a stabilizing MPC at the high level and a shrinking horizon MPC at the low level is devised.

#### A. Stabilizing MPC at the high level

In order to design the high-level regulator in the slow time scale, first define the time index $k \in \mathbb{N}$ associated with a fixed positive integer $N$ so that $h = kN$ and denote by $u^{[s]}(k), x^{[s]}(k),$ and $y^{[s]}(k)$ the samplings of $u, x,$ and $y$ (where $* = s$ or $f$, in turn) and by $u^{[N]} = (u^{[s]}_1, u^{[f]}_1), x^{[N]} = (x^{[s]}_1, x^{[f]}_1),$ and $y^{[N]} = (y^{[s]}_1, y^{[f]}_1)$ the samplings of the input, state, and output variables
corresponding to the time scale $k$. Hence, the sampled system of (2) with $N$ period is given as

\[
\Sigma^N: \begin{cases}
  x^N(k+1) = A^N x^N(k) + B^N u^N(k) \\
  y^N(k) = C x^N(k),
\end{cases}
\]

where $A^N = A^N$, $B^N = \sum_{j=0}^{N-1} A^{N-j-1} B$. Notice that, from (2) and (6), if $x(kN) = x^N(k)$ and the control $u(h) = u^N(k) \forall h \in [kN,kN+N)$, it holds that $x(kN+N) = x^N(k+1)$ and $y(kN+N) = y^N(k+1)$.

The following proposition can be stated for $\Sigma^N$:

**Proposition 1**: The pair $(A^N, C)$ is detectable if $(A, C)$ is detectable.

Also, the following assumption about $\Sigma^N$ is assumed to be holding:

**Assumption 2**: The pair $(A^N, B^N)$ is stabilizable.

**Remark 1**: Note that, slightly different from the detectability condition in Proposition 1 to meet the stabilizability requirement of $\Sigma^N$, we need Assumption III-A to be verified a posteriori once the sampling period $N$ is chosen. This is due to the fact that, starting from the stabilizability of $(A, B)$, there is no guarantee the resultant sampled pair $(A^N, B^N)$ is also stabilizable.

A simple example to illustrate this point is as follows: consider a SISO system described by $x(h+1) = -x(h) + u(h)$, which is obviously stabilizable. However, the $N = 2$ period sampled version $x^2(k+1) = (-1)^2 x^2(k) + (-1 + 1) u^2(k) = x^2(k)$ is not stabilizable.

With (6), it is now possible to state the MPC problem at the high level. At each slow time-step $k$ we solve an optimization problem according to receding horizon principle as follows:

\[
\min_{u^N: [k:k+N_t-1]} J_h \quad \text{s.t. (7)}
\]

where

\[
J_h = \frac{1}{N_t} \sum_{i=0}^{N_t-1} \left( \| y^N(k+i) - y_r(k+i) \|_Q^2 + \| u^N(k+i) - u_r(k+i) \|_R^2 \right)
\]

and where $N_t > 0$ is the adopted prediction horizon. The parameters $Q_h \in \mathbb{R}^{p,p}$, $R_h \in \mathbb{R}^{m,m}$ are positive definite and symmetric weighting matrices, while $P_h \in \mathbb{R}^{n,n}$ is computed as the solution to the Lyapunov equation described by

\[
F^T_h P_h F_h - P_h = -(C^T Q_h C + K^T_h R_h K_h)
\]

where matrix $F_h = A^N + B^N K_h$ is Schur stable and $K_h$ is a stabilizing gain matrix.

The optimization problem (7) is performed under the following constraints:

1. the dynamics (6);
2. the input and state constraints

\[
x^N(k+i) \in \mathcal{X}^N \quad u^N(k+i) \in \mathcal{U}^N
\]

3. the terminal state constraint

\[
x^N(k+N_t) \in \mathcal{X}^N
\]

Thanks to Assumption III-A, the set $\mathcal{X}^N$ is chosen as a positively invariant set for system (6) controlled with the stabilizing control law $u^N(k) = K_h (x^N(k) - x_r) + u_r$, satisfying $K_h (\mathcal{X} \oplus \mathcal{X}_f) \subseteq \mathcal{X} \oplus \mathcal{U}$. Let $u^N(k : k+N_t-1|k)$ be the optimal solution to optimization (7). Only the first element $u^N(k)|k) is applied at time instant $k$, then the values of $x^N(k+1|k)$ and $y^N(k+1|k)$ are updated and the optimization is repeated at the next time instant $k+1$ according to receding horizon principle.

**B. Shrinking horizon MPC at the low level**

Assume now to be at a specific fast time instant $h = kN$ such that the high-level optimization problem (7) with cost (5) has been successfully solved. Thus the computed values of the input $u^N(k) = (u^N_r(k), u^N_f(k))$ and the one-step ahead state prediction $x^N(k+1|k)$ are available. Let us focus on the output performance in the fast time scale within the interval $h \in [kN, kN+N)$. Denoting by $\tilde{y}(h) = (\tilde{y}_r(h), \tilde{y}_f(h))$ the output resulting from (2) with $u(h) = u^N(\lfloor h/N \rfloor)$, the component $\tilde{y}_f(h)$ may expect undesired transient due to the use of the long sampling period at the high level.

For this reason, at the low level the overall control action associated with $y_f$ is refined as

\[
u_f(h) = \tilde{u}_f(h) + \delta u_f(h)
\]

where $\tilde{u}_f(h) = u_f^N(\lfloor h/N \rfloor)$, $\delta u_f$ is computed at the low level by a properly defined optimization problem, see (12).

Since $\delta u_f(h)$ could influence the value of $y_r(h)$ in the fast time scale due to possible nonzero coupling terms from $\Sigma_f$ to $\Sigma_r$ (e.g. $A_{sf}$, $B_{sf}$), it is also convenient to allow a further control freedom of $u_r$ leading to the correction as follows:

\[
u_r(h) = \tilde{u}_r(h) + \delta u_r(h)
\]

and where $\tilde{u}_r(h) = u_r^N(\lfloor h/N \rfloor)$, $\delta u_r$ is another decision variable at the low level.

In view of (10), we write dynamics (2) in the form:

\[
\Sigma: \begin{cases}
  x(h+1) = Ax(h) + Bu(h) + B\delta u(h) \\
  y(h) = Cx(h)
\end{cases}
\]

where $\tilde{u} = (\tilde{u}_r, \tilde{u}_f)$, $\delta u = (\delta u_r, \delta u_f)$.

Accordingly, at any fast time instant $h = kN + t$, letting $\tilde{u}(h) = (k+1)N - 1$.

\[
\delta u_i(h) = \sum_{i=0}^{t-1} \delta u(i) + \delta u_r(h) + \delta u_f(h) + \delta u_f(h)
\]

where $\delta u_r(h) = (\delta u_r(h), \delta u_f(h))$, $\delta u_f(h) = (\delta u_f(h))$, for $h \in [kN, kN+N)$.

The optimization problem (12) is performed under the following constraints:

1. the dynamics (11);
(2) the state and input constraints
\[ x(h+j| h) \in \mathcal{X}, \forall j = 0, \ldots, N - t - 1, \quad (14a) \]
\[ u^{[N]}(\lfloor h/N \rfloor) + \delta u(h + j| h) \in \mathcal{U}, \forall j = 0, \ldots, N - t - 1; \quad (14b) \]
(3) the state terminal constraint
\[ x(kN + N|h) = x^{[N]}(k + 1| k) \] (15)

Thanks to the (15), at the high level the state in the next slow time instant \( x^{[N]}(k + 1) \) can be recovered by the predicted value \( x^{[N]}(k + 1| k) \), i.e., \( x^{[N]}(k + 1) = x^{[N]}(k + 1| k) \).

Remark 2: The rationale of choosing signal \( \tilde{y}^*(h) \) as the reference for the low level lies in the fact that \( y_f(h) \) is expected to react promptly to respond to \( \tilde{y}_f(kN + N) \), while \( y_s(h) \) can be controlled to follow the smooth trajectory \( \tilde{y}_s(h) \) generated from the high level.

Remark 3: It is highlighted that the structure of the proposed approach is different from that of the cascade ones, see for instance [29], for the reason that in the cascade algorithm, the computed input from the high level is considered as the output reference to be tracked at the low level, while the proposed schemes utilize the control action computed from the high level, i.e., \( \bar{u}(h) \), to generate the possible reference profile with model (2) in an open-loop fashion.

C. Summary of the D-MPC algorithm

In summary, the main steps for the on-line implementation of the D-MPC algorithm is given in Algorithm 1.

Algorithm 1 On-line implementation of D-MPC

```
initialization
while for any integer \( k \geq 0 \) do
  h1) compute the control \( u^{[N]}(k| k) \) by solving the optimization problem \( 7 \) with \( 8 \) and update \( x^{[N]}(k + 1| k) \)
  h2) generate the open-loop output \( \tilde{y}^*(h) \) from \( 2 \) with \( u(h) = u^{[N]}(\lfloor h/N \rfloor) \), for all \( h \in [kN, kN + N) \)
  for \( h \leftarrow kN \) to \( kN + N - 1 \) do
    11) compute \( \delta u(h| h) \) with optimization problem \( 12 \) and \( 13 \) and apply \( u(h) = u^{[N]}(\lfloor h/N \rfloor) + \delta u(h| h) \) to \( 2 \)
    12) update \( x(h + 1) \) and \( y(h + 1) \)
  end
  h3) \( k \leftarrow k + 1 \)
end
```

The following theoretical results can be stated:

**Theorem 1:** Under Assumption [1] if the initial condition is such that \( x^{[N]}(0) = x(0) \) and \( 7 \) is feasible at \( k = 0 \), then the following results can be stated for the proposed D-MPC control algorithm

1. The feasibility can be guaranteed:
   - for the high-level problem \( 7 \) at all slow time instant \( k \geq 0 \);
   - for the low-level problem \( 12 \) at all fast time instant \( h \geq 0 \).

2. The slow-time scale system \( \Sigma_s^{[N]} \) enjoys the convergence property, i.e., \( \lim_{k \to +\infty} (u^{[N]}(k), x^{[N]}(k), y^{[N]}(k)) = (u_r, x_r, y_r) \).

3. Moreover, for the low-level problem \( 12 \), it holds that \( \lim_{k \to +\infty} \delta u(h) = 0 \). Finally, \( \lim_{h \to +\infty} (u(h), x(h), y(h)) = (u_r, x_r, y_r) \).

Note that, the terminal constraint (15) plays a crucial role for the closed-loop properties of the D-MPC due to the fact that it guarantees \( x^{[N]}(k + 1) = x^{[N]}(k + 1| k) \). However, since the proposed D-MPC control structure is an upper-bottom one, the computed value of \( \tilde{y}_{f}^{[N]}(k) \) at the high level influences the control performance at the low level due to (15). For this reason, the state \( x_f(h) \) associated with \( y_f(h) \) in the basic time scale might not converge to its nominal value faster than \( x_s(h) \) especially for systems that exhibit nonseparable open-loop dynamics. This problem can be properly coped with in the framework of the incremental D-MPC whose details will be given in the following section.

IV. Incremental D-MPC Algorithm

In this section, we design an incremental D-MPC algorithm that includes at the high level an explicit design of the trajectory of “fast” output \( y_f^{[N]} \) such that it reaches the reference value \( y_{f,s} \) or its neighbour promptly in the slow time scale, also an integral action so as to improve robustness in case of model uncertainties, e.g., due to modelling errors. Compared with D-MPC, this version requires a major modification on the high-level MPC formulation, meanwhile preserves the previous design of the fast shrinking horizon MPC at the low level.

A. Design of the incremental D-MPC

In the following we mainly focus on the redesign of the MPC regulator at the high level. As a further attention is paid at this level to design the trajectory of \( y_f^{[N]} \), it is convenient to partition the sampled system \( 6 \) as the one with the structure similar to \( 1 \). To proceed, according to the structures of \( A \) and \( B \) (see \( 2 \)), we first rewrite the matrices \( A^{[N]} \), \( B^{[N]} \) into the following forms

\[
A^{[N]} = \begin{bmatrix} A_{ss}^{[N]} & A_{sf}^{[N]} \\ A_{fs}^{[N]} & A_{ff}^{[N]} \end{bmatrix}, \quad B^{[N]} = \begin{bmatrix} B_{ss}^{[N]} & B_{sf}^{[N]} \\ B_{fs}^{[N]} & B_{ff}^{[N]} \end{bmatrix},
\]

where \( A_{ss}^{[N]} \in \mathbb{R}^{n_s \times n_s}, B_{ss}^{[N]} \in \mathbb{R}^{n_s \times m_s} \).

In view of this, finally system (6) can be partitioned as two interacting subsystems represented by \( \Sigma_s^{[N]} \) and \( \Sigma_f^{[N]} \) described as follows:

\[
\Sigma_s^{[N]} : \begin{cases} x_s^{[N]}(k + 1) = A_{ss}^{[N]} x_s^{[N]}(k) + A_{sf}^{[N]} x_f^{[N]}(k) + B_{ss}^{[N]} u_s^{[N]}(k) + B_{sf}^{[N]} u_f^{[N]}(k) \\ y_s^{[N]}(k) = C_{ss} x_s^{[N]}(k), \end{cases}
\]

\( \Sigma_f^{[N]} : \begin{cases} x_f^{[N]}(k + 1) = A_{sf}^{[N]} x_s^{[N]}(k) + A_{ff}^{[N]} x_f^{[N]}(k) + B_{sf}^{[N]} u_s^{[N]}(k) + B_{ff}^{[N]} u_f^{[N]}(k) \\ y_f^{[N]}(k) = C_{ff} x_f^{[N]}(k), \end{cases} \)

The following assumption about \( \Sigma_f^{[N]} \) is assumed to hold:
Assumption 3: Matrix $C_f B_{ff}^N$ is full rank. With (16), with the goal of guaranteeing satisfactory control performance related to $y_f$ in the basic time scale, it is convenient to enforce that all the future predictions $y_f^n(k), \forall k > 0$ associated with $x_f^n(k)$ are equal to the reference value $y_{f,r}$ at the high level. In this way, the real output $y_f$ resulting from the controller (12) will reach the reference $y_{f,r}$ in only one slow-time step. However, this restriction might cause infeasibility issue in the cases where the reference $y_{f,r}$ is far from its initial value $y_f(0)$, and/or constraints on the control increments are enforced. For this reason, instead of imposing $y_f^n(k) = y_{f,r} \forall k > 0$, we enforce the following relation

$$y_f^n(k) = \tilde{y}_{f,r} \forall k > 0,$$

(17)

where

$$\tilde{y}_{f,r} = y_f(0) + \alpha(k)(y_{f,r} - y_f(0))$$

and where $\alpha(k)$ is defined as an optimization variable that its value is restricted by $0 \leq \alpha(k) \leq 1$ and reaches 1 in finite time steps, i.e.,

$$\left\{ \begin{array}{l}
\alpha(k) = 0, \quad k = 0 \\
0 \leq \alpha(k) < 1, \quad k \in \{1, \ldots, N_\alpha\} \\
\alpha(k) = 1, \quad k \geq N_\alpha
\end{array} \right.$$

(18)

where $N_\alpha$ is a positive integer. Thanks to Assumption 3 from (16), imposing (17) is equivalent to considering the constraint as follows:

$$u_f^n(k) = (C_f B_{ff}^N)^{-1}(\tilde{y}_{f,r} - C_f(A_f s x_f^n(k) + A_f x_f^n(k) + B_f u_f^n(k)))$$

(19)

Under constraint (19), the time steps required for $y_f^n$ being converging to its reference value $y_{f,r}$ can be defined via properly tuning parameter $N_\alpha$.

With a slightly abuse of notation, we denote $\tilde{\Sigma}^N$ the system from (16) by substituting $u_f^n(k)$ with (19), that is

$$\tilde{\Sigma}^N : \left\{ \begin{array}{l}
x_f^n(k+1) = \tilde{A}^N x_f^n(k) + \tilde{B}^N u_f^n(k) + \tilde{B}_{ff}^N \tilde{y}_{f,r} \\
y_f^n(k) = \tilde{C}_s x_f^n(k)
\end{array} \right.$$

(20)

where

$$\tilde{A}^N = \begin{bmatrix} \tilde{A}^N_s & \tilde{A}^N_{sf} \\ \tilde{A}^N_{fs} & \tilde{A}^N_{ff} \end{bmatrix}, \tilde{B}^N = \begin{bmatrix} \tilde{B}^N_s \\ \tilde{B}^N_{fs} \end{bmatrix}, \tilde{B}_{ff}^N = \begin{bmatrix} \tilde{B}_{ff,s}^N \\ \tilde{B}_{ff,fs}^N \end{bmatrix}, \tilde{C}_s = \begin{bmatrix} C_{ss} \ 0 \end{bmatrix}^T,$$

and where

$$\begin{align*}
\tilde{A}^N_s &= A^N_s - B^N_s (C_f B_{ff}^N)^{-1} C_f A^N_s \\
\tilde{A}^N_{sf} &= A^N_{sf} - B^N_{sf} (C_f B_{ff}^N)^{-1} C_f A^N_s \\
\tilde{A}^N_{fs} &= A^N_{fs} - B^N_{fs} (C_f B_{ff}^N)^{-1} C_f A^N_s \\
\tilde{A}^N_{ff} &= A^N_{ff} - B^N_{ff} (C_f B_{ff}^N)^{-1} C_f A^N_s \\
\tilde{B}^N_{ss} &= B^N_{ss} - B^N_{ss} (C_f B_{ff}^N)^{-1} C_f B^N_{fs} \\
\tilde{B}^N_{fs} &= B^N_{fs} - B^N_{fs} (C_f B_{ff}^N)^{-1} C_f B^N_{fs} \\
\tilde{B}_{ff,s}^N &= B^N_{ff,s} (C_f B_{ff}^N)^{-1} \\
\tilde{B}_{ff,fs}^N &= B^N_{ff,fs} (C_f B_{ff}^N)^{-1} 
\end{align*}$$

Assumption 4: The integer $N$ is such that $\tilde{A}^N$ is stable.

With (20), it is reasonable to define a corresponding MPC problem at the high level similar to (12). We highlight that, with this formulation, the closed-loop performance might be hampered due to possible model uncertainties. Also note that, in view of the definition of $\alpha(k)$, the value of $y_f^n$ will reach $y_{f,r}$ as long as the time index $k \geq N_\alpha$. For these reasons, model (20) is reformulated in the corresponding incremental form. In doing so, the effects by slow (or constant) disturbance can be alleviated (or cancelled) and the dependency on $y_f^n$ will be disappeared for $k \geq N_\alpha$. With the model in velocity form, it is possible to define a MPC problem including an integral action with the goal of output offset-free tracking control. In doing so, the closed-loop system is capable to compensate for constant or slow disturbances, see (30). To this end, denoting $\Delta x_f^n(k) = x_f^n(k) - x_f^n(k-1)$, $\Delta u_f^n(k) = u_f^n(k) - u_f^n(k-1)$, $\Delta \alpha(k) = \alpha(k) - \alpha(k-1)$, $\Delta x_f^n(k) = (y_f^n(k), \Delta x_f^n(k))$, from (20) we compute

$$\tilde{\Sigma}^N : \left\{ \begin{array}{l}
\Delta x_f^n(k+1) = \tilde{A}^N \Delta x_f^n(k) + \tilde{B}^N \Delta \alpha(k) + \tilde{B}_{ff}^N \Delta y_f^n(k) \\
\Delta \alpha(k) = \alpha(k-1) + \Delta \alpha(k) \\
\Delta y_f^n(k) = \tilde{C}_s \Delta x_f^n(k)
\end{array} \right.$$

(21)

where

$$\tilde{A}^N = \begin{bmatrix} I & \tilde{C}_s \tilde{A}^N_s \\ 0 & \tilde{A}^N_{ss} \end{bmatrix}, \tilde{B}^N = \begin{bmatrix} \tilde{C} \tilde{B}^N_s \\ \tilde{B}^N_{ss} \end{bmatrix}, \tilde{B}_{ff}^N = \begin{bmatrix} \tilde{C}_s \tilde{B}_{ff,s}^N \\ \tilde{B}_{ff,ss}^N \end{bmatrix}, \tilde{C} = \begin{bmatrix} I & 0 \end{bmatrix}^T.$$

Proposition 2: The pair $(\tilde{A}^N, \tilde{B}_{ff}^N)$ is stabilizable if and only if

- rank $\left( \begin{bmatrix} \tilde{C} \tilde{A}^N \\ \tilde{A}^N - I \end{bmatrix} \tilde{B}_{ff}^N \right)^T = n + p_s,$
- rank $\left( \begin{bmatrix} 2I & 0 \\ \tilde{C} \tilde{A}^N + I \end{bmatrix} \tilde{B}_{ff}^N \right)^T = n + p_s.$

Under proposition 3, it is possible to find a gain matrix $\tilde{K}_{ss}$ such that $\tilde{F}_{ss} = \tilde{A}^N + \tilde{B}_{ff}^N \tilde{K}_{ss}$ is Schur stable.

Note that, the original constraints on $x_f^n(k)$, $\tilde{u}_f^n(k)$, and $\tilde{u}_f^n(k)$ are not compatible with model (21). We are going to show that, along the same line described in (30), it is possible to represent the control and state variables by the state and input variables in the velocity form, that is

$$\begin{align*}
\Delta x_f^n(k) &= \Gamma_{ss} \Delta x_f^n(k+1) \\
\Delta u_f^n(k) &= \Gamma_{ss} \Delta u_f^n(k+1) \\
u_f^n(k) &= (C_f B_{ff}^N)^{-1} \alpha(k)(y_{f,r} - \gamma_{ss} \Delta x_f^n(k+1))
\end{align*}$$

(22)

where

$$\begin{align*}
\Gamma_{ss} &= (C_f B_{ff}^N)^{-1} C_f \tilde{A}_f A_f B_{ff}^N \Gamma_s \\
\Gamma_{ss} &= \begin{bmatrix} \tilde{A}_s & 0 \end{bmatrix} \begin{bmatrix} \tilde{A}^N_{ss} \end{bmatrix}^{-1} \\
\Gamma &= \begin{bmatrix} \tilde{C} \tilde{A}^N_s \\ \tilde{A}^N_{ss} - I_s \end{bmatrix} \begin{bmatrix} \tilde{B}_{ff,s}^N \end{bmatrix}^{-1}.
\end{align*}$$

In view of (3a), (4a), (4b), (22), the constraints to be considered at the high level are as follows:

$$\begin{bmatrix} \Gamma_{ss} & \Gamma_{ss} \\ -\Gamma_{ss} & 0 \end{bmatrix} \begin{bmatrix} \Delta x_f^n(k+1) \\ \Delta y_f^n(k+1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \alpha(k) y_{f,r} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \alpha(k) y_{f,r}$$

(23)
Based on (21) and (23), now it is possible to state the incremental MPC problem at the high level. At each slow time-step \( k \) we solve an optimization problem according to receding horizon principle as follows:

\[
\min_{\Delta u^{[N]}_{\ell}(k+1|k)} J_{\ell},
\]

where

\[
J_{\ell} = \sum_{i=0}^{N_{\ell}-1} \left[ \|\dot{x}^{[N]}(k+i) - \hat{C}^T y_{\alpha}(k+i)\|^2_{\hat{R}} + \|\Delta u^{[N]}_{\ell}(k+i)\|^2_{\hat{R}} \right] + \\
\gamma \|u(k+i)\|^2_{\hat{R}}.
\]

\[
(25)
\]

\( \gamma \) is a positive scalar, \( N_{\ell} \) is the adopted prediction horizon. The positive definite and symmetric weighting matrices \( \hat{Q}_{\ell,\ell} \in \mathbb{R}^{(n+\rho \ell) \times (n+\rho \ell)} \), \( \hat{R}_{\ell,\ell} \in \mathbb{R}^{m_{\ell} \times m_{\ell}} \) are free design parameters, while \( \hat{R} \) is computed as the solution to the Lyapunov equation

\[
\hat{F}_{\ell}^T \hat{R}_{\ell} \hat{F}_{\ell} - \hat{P}_{\ell} = -\left( \hat{Q}_{\ell,\ell} + \hat{R}_{\ell,\ell} \hat{F}_{\ell,\ell} \right).
\]

\[
(26)
\]

The optimization problem (24) is performed under the following constraints:

1. dynamics (21), constraint (18) and (23);
2. the state terminal constraint

\[
\dot{x}^{[N]}(k+N_{\ell}) \in \mathcal{S}_{\ell},
\]

where the set \( \mathcal{S}_{\ell} \) is a positively set invariant for the nominal system of (21), i.e.,

\[
\Delta \Sigma^{[N]} : \begin{cases} \\
\dot{x}^{[N]}(k+1) = \hat{A}^{[N]} \dot{x}^{[N]}(k) + \hat{B}^{[N]} \Delta u^{[N]}_{\ell}(k) \\
\hat{y}^{[N]}_{\ell}(k) = \hat{C} \dot{x}^{[N]}(k),
\end{cases}
\]

\[
(27)
\]

that is controlling the stabilization control law \( \Delta u^{[N]}_{\ell}(k) = \hat{K}_{\ell,\ell}(\dot{x}^{[N]}(k) - \hat{C}^T y_{\alpha}(k)) \) such that \( \hat{F}_{\ell,\ell} \mathcal{S}_{\ell} \subseteq \mathcal{S}_{\ell} \) under constraint (23). Let \( \Delta u^{[N]}_{\ell}(k; k+N_{\ell} - 1|k) \) be the optimal solution to optimization (24). Only the first element \( \Delta u^{[N]}_{\ell}(k|k) \) is applied at time instant \( k \), then the values of \( \dot{x}^{[N]}(k+1|k), \hat{y}^{[N]}_{\ell}(k+1|k) \) are updated. The real input \( u^{[N]}_{\ell}(k) \) at time instant \( k \) is given by \( u^{[N]}_{\ell}(k) = u^{[N]}_{\ell}(k-1) + \Delta u^{[N]}_{\ell}(k|k) \), also from (19) we can compute the value of \( u^{[N]}_{\ell}(k) \). At this time instant, the state \( x^{[N]}(k+1|k) \) is available by applying \( u^{[N]}(k) = (u^{[N]}_{\ell}(k), u^{[N]}_{f}(k)) \) to (6).

With the above available information, the fast MPC problem described in the previous section, i.e., (12) with cost (14) is solved recursively in the fast interval \( [kN, kN+N] \) according to shrinking horizon principle. Then the optimization problem (24) with cost (25) is repeated at \( k+1 \) according to reeding horizon principle.

### B. Summary of the incremental D-MPC algorithm

To better clarify the requirements for the implementation of the incremental D-MPC algorithm and its difference with the D-MPC algorithm, the main steps for the on-line implementation is given in Algorithm 2.

The following theoretical results are stated:

**Theorem 2:** Under Assumptions 1–4 if the initial condition is such that \( x^{[N]}(0) = x(0) \) and \( N_{\alpha} \) is reachable by Algorithm 2 such that (24) is feasible at \( k = 0 \), then the following results can be stated for the incremental D-MPC algorithm (i.e., high-level problem (24) with cost (25) and low-level problem (12) with (13)):

1. The feasibility can be guaranteed:
   - for the high-level problem (24) at all slow time instant \( k > 0 \);
   - for the low-level problem (12) at all fast time instant \( h \geq 0 \).

2. The slow-time scale system (21) enjoys the convergence property, i.e., \( \lim_{k \rightarrow +\infty} (x^{[N]}(k), \Delta u^{[N]}_{\ell}(k)) = (x_r, u_r) \). Consequently, \( \lim_{k \rightarrow +\infty} (x^{[N]}(k), u^{[N]}_{f}(k)) = (x_r, u_r) \).

3. For the low-level problem (12) it holds that \( \lim_{k \rightarrow +\infty} \delta u(h) = 0 \). Finally, \( \lim_{h \rightarrow +\infty} (u(h), x(h), y(h)) = (u_r, x_r, y_r) \).

### V. Simulation example

The proposed D-MPC and incremental D-MPC algorithms are used for the control of a BT system including the comparisons of their control performances with a traditional decentralized PID controller and a multi-rate MPC.

#### A. Description of the BT system and its linearized model

A 160 MV BT system in [31] is considered and its dynamic diagram is presented in Fig. 2. The input variables applied to the boiler are the fuel flow \( q_f \) (kg/s) and feedwater flow \( q_w \) (kg/s), while the controlled variables of the boiler are drum pressure \( P \) (kg/cm²) and water level \( L \) (m). The control and controlled variables of the turbine are the steam control \( q_s \) (kg/s) and the electrical power output \( Q \) (MV). Typically, the goal of BT control is to regulate the electrical power to meet the load demand profile meanwhile to minimize the variations of internal variables such as water level and drum pressure within their safe sets. Moreover, drum pressure must...
also be controlled properly in the operation range to respond to possible turbine speed changes caused by load demand variations. Many works have been addressed at this point that focus on deriving satisfactory closed-loop control performance of electrical power, see e.g. [32], [33], [34]. In this scenario, the control related to the output variables such as electrical power and drum pressure is a major issue that must be tackled properly to respond to frequent load demand variations, while the water level can be adjusted smoothly under its constraint with the possibility to follow its desired value. This makes it reasonable in this case to apply the proposed dual-level control algorithms. The continuous nonlinear dynamic model described in [31] is given by

\[
\begin{align*}
\dot{\rho} &= (141q_w - (1.1q_s - 0.19)P)/85 \\
\dot{P} &= -0.0018q_f P^{9/8} + 0.9q_f - 0.15q_w \\
\dot{Q} &= (0.073q_s - 0.016)P^{9/8} - 0.1Q
\end{align*}
\]

where \( \rho \) is the fluid density (kg/cm\(^3\)). The control variables are limited by 0 ≤ \( q_f, q_w, q_s \) ≤ 1 and their rate constraints are also considered, i.e., -0.007 ≤ \( \dot{q}_f \) ≤ 0.007, -2 ≤ \( q_s \) ≤ 0.2, -0.05 ≤ \( q_w \) ≤ 0.05. The water level relies on a static nonlinear mapping from the state and input variables in \( \Sigma \). For simplicity, instead of water level, the fluid density is selected as a controlled variable. The considered operation point is \((\rho_r, P_r, Q_r) = (513.6, 129.6, 105.8)\). \((q_{w,r}, q_{f,r}, q_{s,r}) = (0.663, 0.505, 0.828)\). The continuous linearized model at this operation point is computed and described by

\[
\begin{align*}
\dot{x} &= A_c x + B_c u \\
y &= C x,
\end{align*}
\]

where

\[
A_c = \begin{bmatrix} 0 & -0.008 & 0 \\ 0 & -0.003 & 0 \\ 0 & 0.092 & -0.1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 1.66 & 0 & -1.68 \\ -0.15 & 0.9 & -0.43 \\ 0 & 0 & 17.4 \end{bmatrix},
\]

\( C = I \), the state and output variables are \( y = x = (\rho - \rho_r, P - P_r, Q - Q_r) \), while the input variables are \( u = (q_w - q_{w,r}, q_f - q_{f,r}, q_s - q_{s,r}) \). The unitary step response of \( \Sigma \) is presented in Figure 3 which displays that the system outputs are strongly coupled, and the dynamics is not strictly separable.

\[ \text{Fig. 2. Diagram of the BT dynamics.} \]

\[ \text{Fig. 3. Unitary impulse response of the BT dynamics.} \]

\[ \text{Fig. 4. Control variables of the BT plant: red continuous lines (blue dot-dashed lines) represent the inputs computed with the D-MPC (incremental D-MPC) approach, while black dashed lines are the ones computed with the decentralized PIDs.} \]

\[ \text{B. Devising the D-MPC and incremental D-MPC regulators} \]

In order to implement the proposed dual-level control algorithms, the system’s continuous-time model \( \Sigma \) has been sampled with \( \Delta t = 1s \) to obtain the discrete-time counterpart in the fast time scale. The resulting system has been rewritten to derive model \( \Sigma_f \), where the input, state, and output variables associated with \( \Sigma_f \) to be controlled smoothly are chosen as \( u_f = q_w - q_{w,r}, \) \( x_f = \rho - \rho_r, \) and \( y_f = x_f \) while the corresponding ones associated with \( \Sigma_f \) to be controlled in a prompt fashion are \( u_f = (q_f - q_{f,r}, q_s - q_{s,r}) \), \( x_f = (P - P_r, Q - Q_r) \), and \( y_f = x_f \). The resulting model has been re-sampled with \( N = 20 \) to obtain \( \Sigma_f \) and \( \Sigma_{f_1} \) to be used at the high level.

1) Design of the D-MPC regulator:

- The high-level stabilizing MPC \( \Sigma_f \) with cost \( \Sigma_{f_1} \) has been implemented with \( Q_H = I \) and \( K_H = \text{diag}(2, 20, 20) \), and prediction horizon \( N_H = 20 \). The control gain matrix \( K_H \)
is selected by solving an infinite horizon LQ problem. The terminal penalty is computed according to (9) and the solution is

\[ P_{H} = \begin{bmatrix} 10.34 & 2.54 & 0.65 \\ 2.54 & 59.19 & 4.73 \\ 0.65 & 4.73 & 8.01 \end{bmatrix}. \]

The terminal set has been chosen according to the algorithm described in [35], i.e., \( \mathcal{X}_{T} = \{x(x-x_{0})^TP_{H}(x-x_{0})\} \leq 0.221 \).

- The low-level shrinking horizon MPC (12) with cost (13) has been designed with \( Q = I \) and \( R = \text{diag}(1, 1, 10) \).

2) Design of the incremental D-MPC regulator:

- The high-level stabilizing MPC (24) with cost (25) has been implemented with \( N_{d} = 2 \) (see Algorithm 3). \( Q_{H} = I \) and \( R_{H} = \text{diag}(20, 20) \), and prediction horizon \( N_{H} = 20 \). The control gain matrix \( \mathcal{K}_{sH} \) is selected by solving the corresponding infinite horizon LQ problem. The terminal penalty is computed according to (9) and the solution is

\[ P_{H} = \begin{bmatrix} 5.89 & -1.39 & -0.0001 & -0.0002 \\ -1.39 & 7.2 & -0.0002 & 0.0003 \\ -0.0001 & -0.0002 & 5 & 0 \\ -0.0002 & 0.0003 & 0 & 5 \end{bmatrix}. \]

The terminal set has been chosen according to the algorithm described in [35], i.e., \( \mathcal{X}_{T} = \{x(x-x_{0})^TP_{H}(x-x_{0})\} \leq 0.269 \).

- The low-level shrinking horizon MPC (12) with cost (13) has been designed with \( Q = I \) and \( R = \text{diag}(1, 1, 10) \).

3) Simulation results: The proposed dual-level control algorithms have been applied to the linear BT system by solving an output reference tracking problem in the basic time scale. The output set-point \( y_{r} = (10, 1, -2) \) is initially considered; while at time \( t = 400 \) s, due to load variation, the reference value has been reset according to the new load profile, i.e., \( y_{r} = (5, 2, 4) \). The dual-level control algorithms have been implemented from null initial conditions. In the following, the simulation results have been reported including the comparisons with a traditional decentralized PID controller and the multirate MPC described in [28].

**Comparison with the traditional decentralized PID controller:** The proposed algorithms have been firstly compared with decentralized continuous PID controller. The decentralized PIDs, one for each input/output pair, have been designed with all the selected tuning parameters listed in Tab. [II]. The simulation results have been reported in Fig. [45] which show that, after an initial transient, inputs and outputs return to their nominal values, until the change of the reference occurs when the dual-level and decentralized PID control systems properly react to bring the input and output variables to their new steady-state values. Note that the closed-loop dynamics of the three approaches associated with input/output pair \( (u_{i}, y_{i}) \) are comparable, while the corresponding control performances of the input/output pair \( (u_{f}, y_{f}) \) are significantly different. To be specific, the pairs \( (u_{f}, y_{f}) \) with the proposed D-MPC and incremental D-MPC algorithms react promptly to reference variations that the corresponding \( y_{f} \) can be recovered in about 10 sec and 80 sec respectively; while the one with the PIDs experiences a longer transient period (that is almost 350 sec). This reveals that, compared with the PIDs, the proposed D-MPC and incremental D-MPC show strong points in this respect. Also, the control system with the incremental D-MPC reacts slightly faster to reference variations than the one with the D-MPC especially for the control pair \( (u_{f}(2), y_{f}(2)) \).

**Comparison with the multirate MPC in [28]:** The proposed approaches have also been compared with the multirate MPC described in [28]. Note that, due to the usage of finite impulse response representation, the model used for the multirate MPC must be strictly stable. However, the considered system in this paper has a pole on the unitary circle, see [29]. In order to implement the multirate MPC successfully, a feedback compensator \( u_{c} = k y_{c} + v \) has been used firstly, where \( v \) is defined as an auxiliary control variable, and the feedback gain \( k \) is chosen as \( k = -0.005 \). For fair comparison, the design parameters \( Q_{c} \) and \( R_{c} \) have been selected coincident with the proposed MPC algorithms, i.e., \( Q_{c} = \text{diag}(1, 2, \cdots, 20, \cdots, 20) \), \( R_{c} = 2 \). All the comparative simulation experiments have been implemented within Yalmip toolbox installed in MATLAB environment, see [36], in a Laptop with Intel 8 Core i5-4200U 2.30 GHz running Windows 10 operating system. The average values of the computational time for all the approaches are listed in Table [II].

| Control pair | Proportional (P) | Integral (I) | Derivative (D) |
|--------------|-----------------|-------------|---------------|
| \( u(1) \)  | 0.0019          | 2.10^{-10} | -6.007        |
| \( u(2) \)  | 0.24            | 0.006      | 1             |
| \( u(3) \)  | -0.035          | -4.6.10^{-4}| 0.36          |

TABLE I TUNING PARAMETERS OF THE DECENTRALIZED PIDS
times of the proposed algorithms are slightly greater than that of the multirate MPC. This result is acceptable as the state constraints are considered at both levels and the terminal state constraints are included in \( Q \) and \( H \). The cumulative square tracking errors \( J \) is better, but the residual amplitude of the static tracking errors associated with the lack of compensation term \( y \) of the multirate MPC. Therefore, for fully comparison and analysis, we have also repeated the simulation with a larger penalty value associated with \( y \) for multirate MPC, i.e., \( Q_y(1, 1) = 100 \). Similarly, the penalty value associated with \( y \) of the proposed approaches are changed correspondingly with \( Q_y(1, 1) = Q_h(1, 1) = 100 \). The repeated simulation results of the three approaches are presented in Fig. 6-7. These results reveal that, compared with the previous case, the tracking performance of the pair \( (u_y, y) \) with multirate MPC is better, but the residual amplitude of the static tracking error is still evident. For a numerical comparison, the cumulative square tracking errors \( J_u = \sum_{i=1}^{N_{sim}} \| y_u(i) - y_u,r \|^2 \), and \( J_f = \sum_{i=1}^{N_{sim}} \| y_f(i) - y_f,r \|^2 \), along the simulation steps from 1 to \( N_{sim} = 800 \) are collected for all the three approaches and listed in Tab. III which show that, the proposed algorithms enjoy smaller cumulative square tracking errors than the multirate MPC. Interestingly, the tracking error corresponding to the fast control channel with the incremental D-MPC is better than that with the D-MPC at the price of a slightly larger tracking error in the slow channel.

**TABLE II**

| Algorithm | Opt. solved at \( \bar{h} \) | Av. comp. time (s) |
|-----------|-----------------|-----------------|
| Incremental D-MPC | \( h = 2\bar{h}_L \) | 0.585 |
| L-MPC | \( \bar{h} \) | 0.246 |
| D-MPC | \( h = 2\bar{h}_L \) | 0.307 |
| L-MPC | \( \bar{h} \) | 0.240 |
| Multirate MPC | \( h = 2\bar{h}_L \) | 0.239 |
| E-MPC | \( \bar{h} \) | 0.190 |

**TABLE III**

| Approach | \( Q_y(1, 1) = 1 \) | \( Q_y(1, 1) = 100 \) |
|----------|-----------------|-----------------|
| D-MPC | \( 2.36 \times 10^4 \) | \( 2.36 \times 10^4 \) |
| L-MPC | \( 274.8 \) | \( 274.5 \) |
| Incremental D-MPC | \( 2.85 \times 10^4 \) | \( 157.8 \) |
| L-MPC | \( 157.8 \) | \( 158.0 \) |
| Multirate MPC | \( 8.04 \times 10^4 \) | \( 3.0 \times 10^4 \) |
| \( 1.29 \times 10^4 \) | \( 1.29 \times 10^4 \) |

**VI. CONCLUSION**

In this paper, two dual-level MPC control algorithms have been proposed for linear multi-timescale systems subject to input and output constraints. The proposed MPC algorithms rely on clearly time separation, so allow to deal with control problems in different channels. In view of their
main properties, the proposed algorithms are, based on the solution based on MPC with dual-level structure, suitable not only to cope with control of singularly perturbed systems but also to impose different closed-loop dynamical performance for systems with nonseparable openloop dynamics. Their recursive feasibility and convergence properties have been proven under mild assumptions. The simulation results concerning the use of the proposed approaches for the control of a BT system including their comparisons with a traditional decentralized PID regulator and a multirate MPC are reported, which show that the proposed algorithms are both effective in generating significantly different closed-loop control behaviors to the output variables. In this respect, the proposed D-MPC has shown an advantageous feature with respect to the PIDs and the multirate MPC, while the incremental D-MPC enjoys better tracking performance than the D-MPC in the fast channel at the price of a slightly larger tracking error in the slow channel.

VII. ACKNOWLEDGMENT

The first author wish to thank Prof. Riccardo Scattolini and Prof. Marcello Farina from Politecnico di Milano, for their fruitful discussions.

APPENDIX A

A. Proof of Proposition 1

According to PBH detectability rank test, the pair \((A,C)\) is detectable if and only if \(\text{rank}(\lambda I - A - C) = n, \forall \lambda \in \mathbb{C} \) and \(|\lambda| \geq 1\). An equivalent form to this condition is that \(v = 0\) is the unique solution to the following linear equations

\[
\begin{aligned}
Av &= \lambda v \\
Cv &= 0,
\end{aligned}
\]

\(\forall \lambda \in \mathbb{C} \) and \(|\lambda| \geq 1\). From (30), \(v = 0\) is the unique solution to \(\lambda^{i-1}Av = \lambda^i v, Cv = 0, \forall i \in \mathbb{N}_+\), which is \(A^i v = \lambda^i v, Cv = 0, \forall i \in \mathbb{N}_+\). In view of this, recalling that \((A,C)\) is detectable, it holds that \(v = 0\) is the only solution to

\[
\begin{aligned}
A^{\mathbb{N}} v &= \mu v \\
Cv &= 0,
\end{aligned}
\]

where \(\mu = \lambda^N\), which implies \((A^{\mathbb{N}}, C)\) is observable for all the modes that their poles \(|\lambda| \geq 1\). Hence, Proposition 1 holds.

B. Proof of Theorem 1

1) Recursive feasibility of the D-MPC (i.e., high-level problem (11) and low-level problem (12)): As the problem (11) is assumed to be feasible at time \(k = 0\), with resorting to Mathematical Induction technique, one can prove the closed-loop recursive feasibility by verifying that if (7) is feasible at any time \(k\), then

(i) the low-level problem (12) is feasible at any fast time \(h \in [kN, kN + N)\); 
(ii) also the high-level problem (7) is feasible at the subsequent slow time instant \(k + 1\).

First, we show that condition (i) can be verified. To proceed, recalling that the high-level problem (7) is feasible at time \(k\), from initial condition \(x(0) = x^{\mathbb{N}}(0)\), it holds that \(x(kN) = x^{\mathbb{N}}(k)\). In view of this and of (15), it is easy to see that the null input sequence \(\delta u(kN : (k + 1)N - 1) = 0\) is a feasible solution to problem (12) at time \(h = kN\). In other words, the feasibility of the low-level problem (12) is guaranteed at the fast time instant \(h = kN\). Based on this, we assume problem (12) is feasible at any time \(h \in [kN, kN + N)\).
and let \( \{ \delta u(h) | h \} \) be its optimal solution. Only the first element \( \delta u(h) \) is applied at the current time \( h \) and the remaining control sequence \( \{ \delta u(h+1), \delta u(h+2), \ldots, \delta u(h+|N|-1) \} \) is a feasible choice at the subsequent time \( h+1 \) according to shrinking horizon principle. Hence, (12) is feasible for any fast time \( h \in [kN, kN+N) \).

As for (ii), assume that at any slow time instant \( k \), the optimal control sequence of (7) can be found, i.e., \( u^*_{\delta}(k) = (u^{N\delta}_{\delta}(k), \ldots, u^{1\delta}_{\delta}(k+N-1)) \) such that \( x^{N}(k+N-1) \in \mathcal{E}_F \). In view of the definition of \( \mathcal{E}_F \) and of the gain matrix \( K_0 \), and thanks to constraint (15), the input sequence \( u^*_{\delta}(k+1: k+N-1+1) = (u^{N\delta}_{\delta}(k+1), \ldots, u^{1\delta}_{\delta}(k+N-1)) \) is a feasible choice at the next instant time \( k+1 \), since the terminal constraint \( x(k+N-1+1) \in \mathcal{E}_F \) is verified. Hence, the recursive feasibility of (7) follows.

2) Convergence of the high-level problem (7): Denote by \( J^*_h(x^{N}(k+1|k)) \) the optimal cost associated with \( u^*_{\delta}(k) = (u^{N\delta}_{\delta}(k), \ldots, u^{1\delta}_{\delta}(k+N-1)) \) at time \( k \) and by \( J^*_h(x^{N}(k+1|k)) \) the suboptimal cost associated with \( u^*_{\delta}(k+1: k+N-1+1) \) at time \( k+1 \). It is possible to write

\[
J^*_h(x^{N}(k+1|k)) - J^*_h(x^{N}(k)) = - (||y^N(k)-y_r||^2_{Q_h} + ||u^*_{\delta}(k)||^2_{R_h}) + ||y^N(k+1|k)-y_r||^2_{Q_h} + ||u^*_{\delta}(k+1|k)||^2_{R_h} \]

In view of (9) and from (31), one has

\[
J^*_h(x^{N}(k+1|k)) - J^*_h(x^{N}(k)) = - (||y^N(k)-y_r||^2_{Q_h} + ||u^*_{\delta}(k)||^2_{R_h}) .
\]

Recalling the fact that \( J^*_h(x^{N}(k+1|k)) \leq J^*_h(x^{N}(k)) \), then

\[
J^*_h(x^{N}(k+1|k)) - J^*_h(x^{N}(k)) \leq - (||y^N(k)-y_r||^2_{Q_h} + ||u^*_{\delta}(k)||^2_{R_h}) .
\]

which implies that \( J^*_h(x^{N}(k+1|k)) - J^*_h(x^{N}(k)) \) converges to zero. Moreover, from (32), one has \( J^*_h(x^{N}(k+1|k)) - J^*_h(x^{N}(k)) \geq ||y^N(k)-y_r||^2_{Q_h} + ||u^*_{\delta}(k+1|k)||^2_{R_h} \) and \( y^N(k) \to y_r \) as \( k \to \infty \). Recalling the definitions of \( Q_h \) and \( R_h \), one has \( \lim_{k \to \infty} y^N(k) = y_r \) and \( \lim_{k \to \infty} u^*_{\delta}(k) = u_r \). In view of Proposition 1, consequently \( \lim_{k \to \infty} x^{N}(k) = x_r \).

3) Convergence of the low-level problem (12): Assume that the high-level system variables have reached their reference values, i.e., \( u^N(k) = u_r \), \( y^N(k) = y_r \). Define \( \delta x(k) = x(kN) - x_r \) and \( \delta y(k) = y(kN) - y_r \). Along the same line as described in (15), in view of dynamics (11) at time instant \( h = kN \), the low-level dynamics at the slow time scale is defined

\[
\delta x(k+1) = A^N \delta x(k) + w(k)
\]

\[
\delta y(k) = C^N \delta x(k),
\]

where \( w(k) = \sum_{j=0}^{N-1} A^{N-j-1} B \delta u(kN + j) \). Since \( \delta x(k) = 0, \forall k \geq 0 \) (due to (15)), it holds that \( w(k) = 0 \). In view of the cost function at the low level, the null sequence \( \delta u(h) : (k+1)N-1 = 0 \) solves the problem (7), which implies that \( \lim_{h \to \infty} \delta u(h) = 0 \) and \( \lim_{h \to \infty} u(h) = u_r \). Finally, \( \lim_{h \to \infty} y(h) = y_r \) and \( \lim_{h \to \infty} x(h) = x_r \). \( \square \)

C. Proof of proposition 2

According to PBH stabilizability rank test, the pair \((\tilde{A}^N, \tilde{B}^N)\) is stabilizable if and only if \( \text{rank}(\begin{bmatrix} \lambda I - \tilde{A}^N \ 
\tilde{C} \tilde{A}^N 
\tilde{B}^N \end{bmatrix}) = n + p_s \), for \( \lambda \in \mathbb{C} \) and \( |\lambda| \geq 1 \). An equivalent form to this condition is that \( v = 0 \) is the unique solution to the following linear equations

\[
\begin{align*}
(\tilde{A}^N) v &= \lambda v \\
(\tilde{B}^N) v &= 0,
\end{align*}
\]

where \( \lambda \in \mathbb{C} \) and \( |\lambda| \geq 1 \).

In view of (21), it is possible to write (34) in the form

\[
\begin{bmatrix} I - \lambda \tilde{A}^N \\
\tilde{C} \tilde{A}^N - \lambda \tilde{I} \\
\tilde{C} \tilde{B}^N \end{bmatrix} v = 0
\]

Since \( \tilde{A}^N \) is stable by Assumption 3, it is obvious to see that for \( |\lambda| > 1 \), \( v = 0 \) is the unique solution to (35).

For \( \lambda = 1 \), \( v = 0 \) is the unique solution to (35) if and only if

\[
\text{rank}(\begin{bmatrix} \tilde{C} \tilde{A}^N \\
\tilde{A}^N \\
\tilde{B}^N \end{bmatrix}) = n + p_s
\]

As for \( \lambda = -1 \), \( v = 0 \) is the unique solution to (35) if and only if

\[
\text{rank}(\begin{bmatrix} 2I \\
\tilde{C} \tilde{A}^N \\
\tilde{A}^N + I \\
\tilde{C} \tilde{B}^N \\
\tilde{B}^N \end{bmatrix}) = n + p_s
\]

\( \square \)

D. Proof of Theorem 2

1) Recusrsive feasibility of the incremental D-MPC (i.e., high-level problem (24) and low-level problem (12)): As \( N_\alpha \) is assumed to be reachable by Algorithm 2 such that (24) is feasible at time \( k = 0 \), along the same line of Section A-B the closed-loop system is recursively feasible as long as the following statement is verified: if the high-level problem (7) is feasible at any time \( k \), then

(i) the low-level problem (12) is feasible at any fast time \( h \in [kN, kN+N) \);

(ii) also the high-level problem (24) is feasible at the subsequent slow time instant \( k+1 \).

The proof of condition (i) is similar to Section A-B. As for (ii), assume that at time instant \( k \) the optimal control sequence \( \{ \} \) is found, i.e., \( \bar{\Delta}u^N(k) = (\delta u^N(k), \ldots, \delta u^N(k+N-1)) \) such that \( \bar{x}^{N}(k+N-1|k) \in \mathcal{E}_F \). Noting the fact that \( N_\alpha \geq N_\alpha \), one has \( \alpha(h+N_\alpha) = 1/\forall k \geq 0 \). In view of this, the input sequence \( \bar{\delta}u^N(k+1: k+N_\alpha-1+1) = (\delta u^N(k+1), \ldots, \delta u^N(k+N_\alpha-1, k+N_\alpha-1|k), \bar{x}^{N}(k+N_\alpha-1|k) \in \mathcal{E}_F \) is a feasible choice at the next time instant \( k+1 \) such that \( \bar{x}(k+N_\alpha-1|k+1) \in \mathcal{E}_F \) can also be verified. Hence, the recursive feasibility of (24) follows.
2) Convergence of the incremental D-MPC: In view of (18) and recalling the feasibility result of (24), along the same line of Section A-B one can compute
\[
\begin{align*}
J_k^H(\bar{x}(k+1)k) - J_k^H(\bar{x}(k)k) & \leq -\left(\|x(k) - C^T y_{r,k}\|_{Q_{k,H}}^2 + \|\Delta_{k,\rho}(kk)\|_{R_{k,H}}^2\right)
\end{align*}
\]
where \(J_k^H\) is the optimal cost. (36) implies that \(J_k^H(\bar{x}(k+1)k) - J_k^H(\bar{x}(k)k)\) converges to zero. Consequently, it holds that \(\|x(k) - C^T y_{r,k}\|_{Q_{k,H}}^2 + \|\Delta_{k,\rho}(kk)\|_{R_{k,H}}^2 \to 0\) as well. Recalling the definitions of \(\tilde{Q}_H\) and \(\tilde{R}_H\), it holds that \(\lim_{k \to \infty} x(k) = C^T y_{r,k}\) and \(\lim_{k \to \infty} \Delta_{k,\rho}(kk) = 0\). Consequently, one has \(\lim_{k \to \infty} y(k) = y_r\), \(\lim_{k \to \infty} u(k) = u_r\). The arguments for the results \(\lim_{k \to \infty} \Delta_t(h) = 0\), \(\lim_{k \to \infty} y(h) = y_r\), \(\lim_{k \to \infty} u(k) = u_r\) are similar to Section A-B □

REFERENCES

[1] D. Naidu, “Singular perturbations and time scales in control theory and applications: an overview,” Dynamic Systems and Control Theory, no. 3, pp. 1–18, 2017.
[2] P. Kokotović, H. K. Khalil, and J. O’reilly, “Singular perturbation methodology in control systems,” 1999.
[3] E. Mishchenko, “Asymptotic methods in singularly perturbed systems,” Consultants Bureaure, 1994.
[4] D. Naidu, “Singular perturbation methods in control systems,” 1998, no. 34.
[5] J. Rodriguez-Vasquez, R. Perez, J. Moriano, and J. Gonzalez, “Advanced control system of the steam pressure in a fire-tube boiler,” IFAC Proceedings Volumes, vol. 41, no. 2, pp. 11 028–11 033, 2008.
[6] M. G. Forbes, R. Patwardhan, H. Hamadah, and R. Gopalani, “Model predictive control in industry: Challenges and opportunities,” IFAC, 2008, no. 8, 2015.
[7] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” Control Engineering Practice, vol. 11, no. 7, pp. 733–764, 2003.
[8] M. Li, P. Zhou, H. Wang, and T. Chai, “Nonlinear multibjective MPC-based optimal operation of a high consistency refining system in papermaking,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, no. 99, pp. 1–18, 2017.
[9] Q. Jin, S. Wu, and R. Zhang, “Improved constrained model predictive control tracking control for networked coke furnace systems over uncertainty and communication loss,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, no. 99, pp. 1–8, 2018.
[10] S. Roshany-Yamchi, M. Cywinski, S. R. Negenborn, B. De Schutter, K. Delaney, and J. Connell, “Kalman filter-based distributed predictive control of large-scale multi-rate systems: Application to power networks,” IEEE Transactions on Control Systems Technology, vol. 21, no. 1, pp. 27–39, 2013.
[11] Y. Zhou et al., “A real-time and fully distributed approach to motion planning for multirobot systems,” 2013.
[12] F. Santos, M. A. Garratt, S. G. Anvarati, and I. Petersen, “Robust hybrid nonlinear control systems for the dynamics of a quadcopter drone,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, no. 99, pp. 1–13, 2018.
[13] Z. Zhou, B. De Schutter, S. Lin, and Y. Xi, “Two-level hierarchical model-based predictive control for large-scale urban traffic networks,” IEEE Transactions on Control Systems Technology, vol. 25, no. 2, pp. 505–511, 2017.
[14] Q. Fang, J. Wang, Q. Gong, and M. Song, “Thermal-aware energy management of an hpc data center via two-time-scale control,” IEEE Transactions on Industrial Informatics, vol. 13, no. 5, pp. 2260–2269, 2017.
[15] B. Picasso, X. Zhang, and R. Scattolini, “Hierarchical model predictive control of independent systems with joint constraints,” Automatica, vol. 74, pp. 99–106, 2016.
[16] L. Grüne and V. G. Palma, “On the benefit of re-optimization in optimal control under perturbations,” 2013.