Optimal Sub-References for Setpoint Tracking
A Multi-level MPC Approach

Sun, D.; Jamshidnejad, A.; De Schutter, B.H.K.

DOI
10.1016/j.ifacol.2023.10.233

Publication date
2023

Document Version
Final published version

Published in
IFAC-PapersOnLine

Citation (APA)
Sun, D., Jamshidnejad, A., & De Schutter, B. H. K. (2023). Optimal Sub-References for Setpoint Tracking: A Multi-level MPC Approach. *IFAC-PapersOnLine*, *56*(2), 9411-9416.
https://doi.org/10.1016/j.ifacol.2023.10.233

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 9411–9416

Optimal Sub-References for Setpoint Tracking: A Multi-level MPC Approach ⋆

Dingshan Sun * **Anahita Jamshidnejad** ** **Bart De Schutter** *

* *Delft Center for Systems and Control, Delft University of Technology, The Netherlands (e-mail: d.sun-1@tudelft.nl, b.deschutter@tudelft.nl).*
** *Department of Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands (e-mail: a.jamshidnejad@tudelft.nl)*

Abstract: We propose a novel method to improve the convergence performance of model predictive control (MPC) for setpoint tracking, by introducing sub-references within a multi-level MPC structure. In some cases, MPC is implemented with a short prediction horizon due to limited on-line computation capacity, which could lead to deteriorated dynamic performance. The introduced multi-level optimization method can generate proper sub-references for the MPC setpoint tracking problem, and efficiently improve the dynamic performance. In the higher level a specific performance criterion is taken as the objective, while explicit MPC is utilized in the lower level to represent the control input. The generated sub-references are then used in MPC for the real system with prediction horizon restrictions. Setpoint-tracking MPC for linear systems is used to illustrate the approach throughout the paper. Numerical simulations show that MPC with sub-references significantly improves the convergence performance compared with regular MPC with the same prediction horizon. Thus, it can be concluded that MPC with sub-references has a high potential to tackle more complicated control problems with limited computation capacity.

Keywords: Model predictive control, setpoint tracking, sub-reference, multi-level MPC.

1. INTRODUCTION

Model predictive control (MPC) is an online optimization-based control method that traces back to the 1970s (Morari and Lee, 1999; Mayne, 2014). It has been widely used in the process industry due to its conceptual simplicity and its ability to easily and effectively handle complex systems with many inputs and outputs and hard constraints (Qin and Badgwell, 2000, 2003). Typically, only the first element of the optimized control sequence is implemented to the controlled system, and then the prediction horizon is moved forward for one control step. Abundant stability and optimality results have been derived for MPC (Mayne et al., 2000), which means that the theory of this technique is mature.

However, in practice the limited computational capacity may only allow for a short prediction horizon for online MPC optimization, especially for nonlinear MPC due to the complexity of solving a nonlinear non-convex optimization problem. However, a short prediction horizon may lead to deteriorated dynamic performance, since the controller only considers the short-term cost and ignores the long-term effects. Therefore, it is crucial to achieve a balance between computational complexity and performance, by choosing a proper prediction horizon.

Setpoint tracking is an important application of MPC, in both traditional industrial implementations (e.g., chemicals, polymers, and air and gas processing (Qin and Badgwell, 2000, 2003)), and applications in robotics (e.g., control of quadrotors (Watterson and Kumar, 2015; Bouffard et al., 2012), autonomous vehicles (Richter et al., 2018), and humanoid robots (Erez et al., 2013)). A short prediction horizon in MPC setpoint tracking may cause an undesired overshoot of the system trajectory, or even oscillations, which would result in inaccurate tracking and waste of energy. Therefore, such undesired effects should be avoided in practical use.

One known approach that has been introduced to overcome the online computational issue of MPC is explicit MPC (Alessio and Bemporad, 2009). Explicit MPC precomputes the optimal control law offline as a function of all feasible states. The resulting control law for a linear system with a quadratic cost function is known to be piecewise affine (PWA), and hence, the online optimization problem is reduced to a simple function evaluation (Bemporad et al., 2002; Pistikopoulos et al., 2002; Alessio and Bemporad, 2009). However, explicit MPC is mainly suitable for small-scale systems, as for large-scale systems it will lead to high storage space requirements. Several attempts have been made to address such limitations, such as shortening the prediction horizon (Tøndel and Johansen, 2002), reducing the number of state partitions by optimally merging regions where the affine gain is the same (Geyer et al., 2008), or relaxing the Karush–Kuhn–Tucker conditions, such as nonnegativity of the dual variables (Bemporad and Filippi, 2003). Nonetheless, all these methods

⋆ This work is sponsored by China Scholarship Council, Grant No. 201806230254.

10.1016/j.ifacol.2023.10.233

improve the computational efficiency of explicit MPC at the cost of a possible loss of closed-loop performance.

In this paper, we propose a multi-level MPC approach based on sub-references for an MPC setpoint tracking problem with short prediction horizons. The main aim is to improve the convergence performance of MPC without extra on-line computational burden. This paper is organized as follows. Related work and the idea behind the approach are introduced in Section 2, as well as the definition of sub-references. Then the sub-reference MPC problem is formulated as a two-level optimization problem in Section 3. In Section 4 we discuss how to solve the above optimization problem, followed by numerical simulations and discussion of results in Section 5. Finally, conclusions are given in Section 6.

2. SUB-REFERENCES

2.1 Related work

Gilbert et al. (1994) propose a control method called reference governor for a discrete-time linear system with input constraints. The governor is used to attenuate the control input to avoid saturation, especially when the reference command is too large to follow. The generated artificial reference played an important role in guaranteeing stability of the closed-loop system and convergence to the desired reference. A similar idea is adopted by Limón et al. (2008), where the approach is extended to the reference tracking problem for linear MPC. Artificial references are generated based on the considered piecewise constant references, and are added explicitly in the objective function. In the cost function, the deviation of the system state from the artificial reference is penalized, as well as the distance between the artificial reference and the target reference. Recursive feasibility and convergence can be proved, and the attraction domain of the proposed MPC is enlarged. Limón et al. (2012) further extended their results to the MPC tracking problem with periodic references.

Applications of the concept can be found in Klaučo et al. (2017), which introduces a reference governor in MPC for control of a magnetic levitation system. Setpoints are optimized by the MPC-based reference governor for the inner feedback control loop. The formulated optimal control problem can be solved efficiently with a parametric optimization technique, and the solution can be expressed as a continuous piecewise affine function similar to explicit MPC. Experimental results show that the proposed method can improve the reference-tracking performance compared to the case without a reference governor. This approach is further used by Holaza et al. (2018) to control a continuous stirred-tank reactor, and applied by Klaučo and Kvasnica (2017) to control a boiler-turbine unit. Simulations or experimental results show the effectiveness of the approach in reference tracking.

2.2 Main idea behind our approach

Unlike the studies mentioned above, we explicitly consider the limited on-line computation capacity and the resulting performance deterioration. To address this issue, the concept of sub-reference is proposed, which is different from the artificial reference generated by a reference governor as in the literature. In our method, the generated sub-references are piecewise constant, which means they do not need to be optimized for each control time step. In addition, sub-references are optimized by a multi-level MPC optimization problem, to minimize a cost function specified for a desired performance index.

The main steps of the proposed approach go as follows. A *nonlinear system* can be simplified or linearized. Then, sub-references can be designed through an optimization procedure. The parameters to be optimized include the values of the sub-references and the switching time instants of the sub-references. The whole optimization procedure contains two levels. The optimization is performed with an alternative cost function and the parameters of sub-references as optimization variables at the high level, and at the low level MPC is used to track the sub-reference at each stage (i.e., the duration of each sub-reference). The high-level problem can be solved using time instant optimization (TIO) and the low-level problem will be solved using explicit MPC (see Section 4 for more details). After the optimization procedure, the generated sub-references are then used in on-line setpoint tracking for the original system.

2.3 Definition of sub-references

Consider a discrete-time linear setpoint regulation MPC problem, in which the following quadratic optimization problem is solved at every control time step $k \in \mathbb{N}$ in a receding horizon fashion:

$$\begin{aligned}
&\min_{\tilde{u}_k} J(\tilde{u}_k, x(k), r, N_\mathrm{p}) = \min_{\tilde{u}_k} \sum_{\kappa=1}^{N_\mathrm{p}} (y_{k+\kappa|k} - r)^\top \cdot \\
&\quad Q(y_{k+\kappa|k} - r) + \sum_{\kappa=0}^{N_\mathrm{p}-1} u_{k+\kappa|k}^\top R u_{k+\kappa|k} \\
&\text{s.t. } x_{k+\kappa+1|k} = A x_{k+\kappa|k} + B u_{k+\kappa|k}, \\
&\quad y_{k+\kappa+1|k} = C x_{k+\kappa+1|k} + D u_{k+\kappa|k}, \kappa = 0, 1, ..., N_\mathrm{p}-1, \\
&\quad E\tilde{x}_k + F\tilde{u}_k \leq G, \\
&\quad x_{k|k} = x(k),\ k \in \mathbb{N}.
\end{aligned} \tag{1}$$

where $x_{k+\kappa|k}$, $y_{k+\kappa|k}$, $u_{k+\kappa|k}$ are the predicted state, output, and input for control time step $k+\kappa$ across the prediction window of N_p control steps on the basis of the information available at control time step k where N_p is called the prediction horizon, $\tilde{u}_k = [u_{k|k}^\top, ..., u_{k+N_\mathrm{p}-1|k}^\top]^\top$, $\tilde{x}_k = [x_{k+1|k}^\top, ..., x_{k+N_\mathrm{p}|k}^\top]^\top$, $Q = Q^\top \succeq 0$, $R = R^\top \succ 0$, A, B, C, D are the system matrices, E, F, G are properly defined matrices that represent the constraints on inputs, states, and output, and r is the target setpoint for the output. For the sake of simplicity, the control horizon often used in MPC is assumed to be N_p as well.

We introduce N sub-references in order to improve the convergence performance of MPC, e.g., to reduce the overshoot without increasing the on-line computational burden. For example, we set the reference as r_0 for all control time steps from k_0 up to $k_1 - 1$, r_1 for all control time steps from k_1 up to $k_2 - 1$. The final sub-reference is set to the setpoint r for all control time steps from k_{N-1}

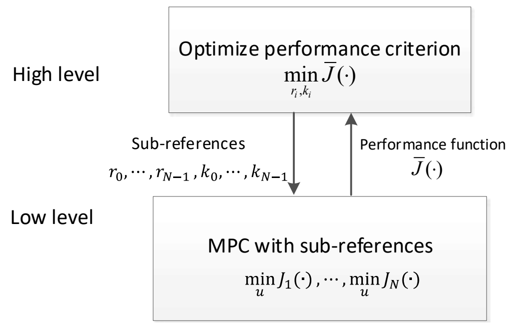

Fig. 1. The structure of two-level optimization of sub-references

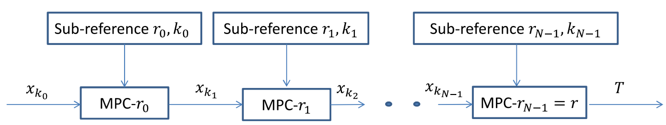

Fig. 2. A sequence of sub-MPC problems at the lower level

up to k_N. Note that $N = 1$ results in the regular setpoint tracking problem. Fig. 3 illustrates the concept of sub-references.

The proposed sub-reference MPC approach works because the sub-references can smoothen the process of convergence, by introducing stage setpoints to steer the outputs to reach the final setpoint gradually. The key point regarding the proposed approach is to find the appropriate sub-reference time steps k_i to introduce the sub-references and the proper magnitudes r_i of the sub-references, for $i = 0, 1, \cdots, N-1$. Next we explain how to find the parameters r_i, k_i, $i = 0, 1, \cdots, N-1$ to optimize a given objective function.

3. PROBLEM FORMULATION

To determine the parameters of the sub-references, a two-level optimization structure, as shown in Fig. 1, is proposed. At the high level, the cost function $\bar{J}$ that represents the simplified or alternative performance criterion, e.g., minimizing the convergence time or overshoot, is optimized to obtain the sub-references parameters r_i, k_i, $i = 0, 1, \cdots, N-1$; meanwhile, at the lower level the original MPC problem can be decomposed into N low-level MPC problems according to the introduced N sub-references, and the sub-MPC problems are solved with regard to their corresponding sub-references and cost functions, in a sequential style along the timeline, as shown in Fig. 2.

The whole procedure can be formulated as the following problem, which is solved for every setpoint r:

$$\min_{r_0,\cdots,r_{N-1},k_0,\cdots,k_{N-1}} \bar{J}(\cdot)$$

$$\text{s.t.} \begin{cases} \min_{\tilde{u}_k} J(U, x(k), r_0, N_\text{p}), \ k \text{ from } k_0 \text{ to } k_1 - 1 \\ \vdots \\ \min_{\tilde{u}_k} J(U, x(k), r_{N-1}, N_\text{p}), \ k \text{ from } k_{N-1} \text{ to } T(\cdot) \\ x_{k+\kappa+1|k} = Ax_{k+\kappa|k} + Bu_{k+\kappa|k}, \\ y_{k+\kappa+1|k} = Cx_{k+\kappa+1|k} + Du_{k+\kappa|k}, \kappa = 0, \ldots, N_\text{p} - 1, \\ E\tilde{x}_k + F\tilde{u}_k \leq G, \end{cases} \quad (2)$$

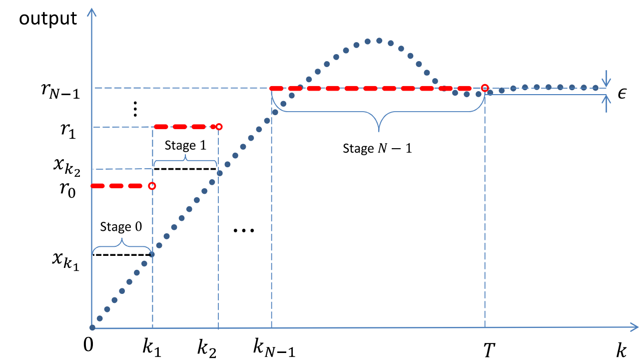

Fig. 3. Illustration of the sub-reference approach; the dotted curve represents the discrete-time trajectory of the output, while the bold dashed lines represent the sub-reference

where k_0 is the starting time step and k_N is the end time step of reference r. The relationship between the parameters as well as the partition of stages are shown in Fig. 3, where the bold dashed lines represent the sub-references in the stages. Problem (2) contains the higher level and the lower level optimization simultaneously, with each sub-optimization problem $\min_{\tilde{u}_k} J(\cdot)$ subject to the system constraints in (1).

The objective function $\bar{J}(\cdot)$ is user-defined to optimize any required performance criteria. As an illustration, let us consider the convergence time $T(\cdot)$, where $T(\cdot)$ is defined as the minimal number of time steps that is required such that the output trajectory converges to the set-point r with a given tolerance ϵ (see Fig. 3). Here ϵ is the convergence tolerance, and the trajectory is assumed to be convergent if it stays within the tolerance bound for N_ϵ time steps. Obviously $T(\cdot)$ is a function of parameters r_i, k_i, $i = 0, 1, ..., N-1$ and ϵ, and it should be represented as $T(r_0, \cdots, r_{N-1}, k_0, \cdots, k_{N-1}, \epsilon)$. Next the problem is how to formulate the objective $\bar{J}(\cdot)$ as a function of the parameters r_i, t_i, for $i = 0, ..., N-1$. The issue is that in general (2) cannot be solved analytically, because finding an explicit relationship between $\bar{J}(\cdot)$ and $r_i, k_i, i = 0, 1, \cdots, N-1$ is in most cases impossible.

Remark 1. The low-level sub-MPC problems already tackle the references inherently, while ensuring state and input constraints. Therefore, the high-level MPC can handle extra performance criteria, such as convergence time or accumulated energy consumption.

4. METHODOLOGY FOR SOLVING THE TWO-LEVEL PROBLEM

In order to solve problem (2), we need to first solve the low-level sub-MPC problems, and then move to the high-level to optimize the sub-reference parameters.

4.1 Explicit controller for lower-level optimization

To tackle the lower-level problem, we propose to consider an explicit expression for the control inputs for every control time step of stage i:

$$u_i(k) = f(x(k), r_i), \quad i = 0, ..., N-1 \tag{3}$$

where $f(\cdot)$ expresses the mapping from the current state $x(k)$ and the current sub-reference r_i to the MPC control inputs for the current control time step k. To formulate the controller as a function of the current state and the parameters $r_i, i = 0, 1, \cdots, N-1$ from (1), we can apply explicit MPC (Bemporad et al., 2002; Pistikopoulos et al., 2002). For the linear case, the explicit formulation of the MPC input results in a piecewise affine (PWA) function, which can be derived through solving a multi-parametric quadratic programming (mp-QP) problem (Alessio and Bemporad, 2009). The derivation of explicit MPC for reference tracking problem can be obtained by following the references, and it is omitted here.

Remark 2. For nonlinear systems, explicit formulation of the input can also be established for the linearized version of the system. Alternatively, the explicit model predictive control law can be approximated by neural networks (Chen et al., 2018; Åkesson and Toivonen, 2006), thus providing the possibility to apply sub-references in a nonlinear system directly. With the explicit controller, the lower-level problem can then be replaced by simple calculation of the state trajectory.

4.2 Solution approaches for the higher-level problem

We can utilize time-instant optimization (TIO) (van Ekeren et al., 2013) to solve the higher-level problem. In this setting, $k_1, \cdots, k_{N-1}$ are the time steps at which the set-point changes. The main feature of TIO is that the time instants and the sub-reference levels are the variables to be optimized (De Schutter and De Moor, 1998; Sadowska et al., 2015), which coincides with the aim of our method. The main challenge is to formulate the relationship between the cost function $\bar{J}(\cdot)$ and the parameters $r_0, \cdots, r_{N-1}, k_0, \cdots, k_{N-1}$. The procedures to obtain sub-references are presented in Algorithm 1.

Algorithm 1 Multi-level MPC for sub-references

1: Given initial state x_0, system dynamics, and the tracking setpoint r
2: Given the number of sub-references N
3: **for** stage i from 0 to $N-1$ **do**
4: Formulate the explicit control input (3)
5: Evolve system states using initial state $x(k_i)$, sub-reference r_i, k_i, and control input (3)
6: Calculate the terminal state $x(k_{i+1})$
7: **end for**
8: Calculate the objective function $\bar{J}(\cdot)$ for the given state trajectory
9: Solve the high-level problem (2) to obtain the sub-references for setpoint r

In order to solve problem (2), we can utilize numerical algorithms, such as a multi-run genetic algorithm (GA) (Whitley, 1994), multi-start sequential quadratic programming (SQP) (Boggs and Tolle, 1995), and so on. In general, more sub-references allow for more flexibility to shape the trajectory, resulting in a better performance; however, this also introduces more parameters within the optimization problem, and increases the computational complexity. Note that this optimization problem can be solved in advance, with the information of the linearized systems, initial states, and target reference. In the next section, it is illustrated via simulation how to reach a balance between optimality and computation efficiency by choosing a proper value for N.

Remark 3. The complexity of problem (2) depends on the definition of objective function $\bar{J}(\cdot)$. Fortunately, the piecewise constant sub-references do not need to be optimized every time step, thus allowing for more time to solve the optimization problem (2). In addition, Algorithm 1 can be performed in a moving horizon strategy to deal with changing setpoints.

5. SIMULATIONS FOR A LINEAR MPC EXAMPLE

In this section, we present a case study on a linear MPC example to illustrate and validate our sub-reference method.

5.1 Setup

Here we take a discrete-time linear system as an example to show the effectiveness of sub-reference MPC. Consider the following system:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu, \\ y &= Cx \end{aligned} \tag{4}$$

where

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \; C = [1 \;\; 0]$$

The initial state is taken as $x(0) = [-10, 0]^\top$; the lower and upper bounds for states and outputs are $x_{\min} = [-15, -15]^\top$, $x_{\max} = [15, 15]^\top$, $y_{\min} = -15$, $y_{\max} = 15$; the lower and upper bounds for the input is $u_{\min} = -1$, $u_{\max} = 1$. To validate the proposed algorithm, a short prediction horizon is chosen $N_{\rm p} = 2$. The penalty matrices for the output and input are $Q = 1, R = 1$, respectively. A series of changing setpoints are considered in this case study.

For each setpoint, we determine 3 sub-references with $N = 3$ by introducing parameters $k_1, k_2, r_0, r_1, r_2 = r$. In order to find the optimal parameters r_0, r_1, k_1, k_2, the explicit controller (3) can be calculated by using the MPT3 toolbox in Matlab (Herceg et al., 2013). After the implementation of explicit MPC in MPT3, the exact formulation of (3) can be obtained:

$$u_{i,j}(k) = H_j \theta_i(k) + g_j \text{ for } K_j \theta_i(k) \le b_j \tag{5}$$

where $\theta_i(k) = [x^\top(k), r_i]^\top$ is the vector of parameters, j denotes the index of the corresponding region for the current parameter vector $\theta_i(k)$, and H, g are the matrix and the vector that define the explicit controller while K, b are the matrix and the vector that represent the critical regions. In this case study, the objective function $\bar{J}(\cdot)$ is the convergence time $T(\cdot)$. The convergence tolerance is $\epsilon = 0.1$. With the explicit controller (5), the given initial states and setpoint, and parameters $r_i, k_i, i = 0, 1, 2$, we can calculate the final objective $T(\cdot)$ in a numerical way.

To obtain the optimal parameters of this nonlinear and non-convex function, we use `MultiStart` function of the Matlab Optimization toolbox, in which we choose the SQP algorithm and take 1000 starting points to solve the nonlinear and non-convex optimization problem. Furthermore, simulations of linear MPC with different numbers of sub-references are carried out to compare the various choices of the number of sub-references.

5.2 Results and discussions

Three setpoints are set for tracking (see Fig. 4), which are 5, -5, and 15, respectively. For each setpoint, the sub-references are calculated. The problem (2) can be solved efficiently within a few seconds. The sub-references $r_{i,j}$ for each setpoint are (i represents the setpoint, and j represents the corresponding sub-reference):

$$\begin{cases} r_{1,1} = 3.1991, \text{ for } k \text{ from 0 to 8,} \\ r_{1,2} = 4.7153, \text{ for } k \text{ from 8 to 11,} \\ r_{1,3} = 5, \text{ for } k \text{ from 11 onwards,} \\ r_{2,1} = -5.9853, \text{ for } k \text{ from 30 to 33,} \\ r_{2,2} = -4.4616, \text{ for } k \text{ from 33 to 37,} \\ r_{2,3} = -5, \text{ for } k \text{ from 37 onwards,} \\ r_{3,1} = 6.0613, \text{ for } k \text{ from 60 to 61,} \\ r_{3,2} = 11.8442, \text{ for } k \text{ from 61 to 70,} \\ r_{3,3} = 15, \text{ for } k \text{ from 70 onwards.} \end{cases}$$

Note that the convergence time can be less then the switching time instance of the sub-reference. This makes sense because when the system is following the sub-references, the trajectory already converges to the target setpoint and stays around the setpoint within the convergence tolerance in the subsequent time steps.

MPC with a longer prediction horizon is implemented, in which the control horizon $N_c = 2$ and the prediction horizon $N_p = 3$. Thus both MPC approaches have a comparable online computational time. The comparison of the convergence performance is presented in Fig. 4. We see that the overshoot is almost eliminated using the sub-reference MPC approach. The convergence time is reduced significantly compared to MPC without sub-references that has a longer prediction horizon. For the setpoint at the upper boundary of the system, sub-reference MPC can track without violating the constraints, while conventional MPC with a longer horizon cannot avoid exceeding the upper bound. Therefore, it is shown that the sub-reference MPC method improves the convergence performance substantially even with a shorter prediction horizon.

Now, a comparison study is conducted to determine the number of sub-references. For the first setpoint $r = 5$, choosing the number of sub-reference to be $N = 2$, the optimal convergence time is $T = 9$ time steps with the parameters $r_0 = 3.2621, k_1 = 10$. With $N = 4$, the minimum convergence time is $T = 9$ time steps with the parameters $r_0 = 7.6066, r_1 = 6.3460, r_3 = 3.2772, k_1 = 2, k_2 = 4, k_3 = 10$. Similar results are obtained for $N = 3$. The comparison of the trajectories for different sub-references are shown in Fig. 5.

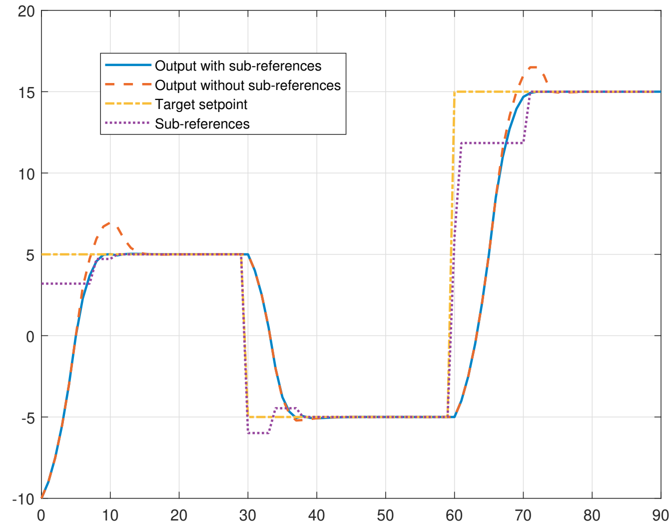

Fig. 4. Comparison of the convergence performance of the closed-loop controlled system for conventional and sub-reference MPC approaches: solid line denotes sub-reference MPC with $N_p = N_c = 2$, dashed line denotes conventional MPC with $N_c = 2, N_p = 3$.

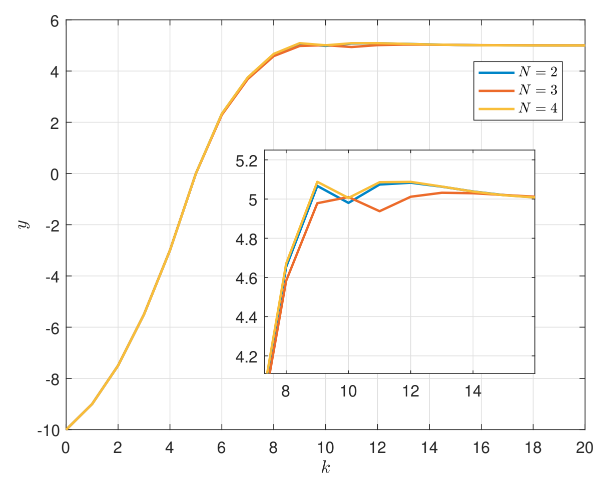

Fig. 5. Comparison of the convergence performance for different numbers of sub-references with details on the convergence point

We see that the performance of sub-reference MPC for different numbers of sub-references is similar, and the same convergence time is achieved for those cases. The mechanism of this method can be summarized as follows. The peak of the realized output trajectory with respect to the sub-reference approaches the final setpoint. Then the sub-reference is switched to the real setpoint and the trajectory can converge to the set-point smoothly and steadily, resulting in a reduction of the overall overshoot. This phenomenon holds for $N = 2, 3, 4$. This implies that a limited number of sub-references can improve the convergence performance substantially, and provides an insight that we can start with a small number of sub-references, e.g., $N = 2$, for the sake of computational efficiency.

6. CONCLUSION AND FUTURE WORK

In this paper we have proposed a multi-level optimization method, called sub-reference MPC, to generate sub-references for MPC setpoint tracking problem. The dynamic performance is considered and optimized explicitly, while the tracking performance is improved even with a short prediction horizon. The effectiveness of the proposed methods is illustrated via numerical simulations. The results show our method outperforms conventional MPC with even a longer horizon, and achieves a better convergence time without violating the constraints when tracking a upper bound reference.

In the future, the generated high-level optimization problem can be further analyzed mathematically, and stability of the method can be investigated. Moreover, a more complexity case study can be conducted on a nonlinear system, and the proposed method can be compared with existing approaches.

REFERENCES

Åkesson, B.M. and Toivonen, H.T. (2006). A neural network model predictive controller. *Journal of Process Control*, 16(9), 937–946.

Alessio, A. and Bemporad, A. (2009). A survey on explicit model predictive control. In *Nonlinear Model Predictive Control*, 345–369. Springer.

Bemporad, A. and Filippi, C. (2003). Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. *Journal of Optimization Theory and Applications*, 117(1), 9–38.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Boggs, P.T. and Tolle, J.W. (1995). Sequential quadratic programming. *Acta Numerica*, 4, 1–51.

Bouffard, P., Aswani, A., and Tomlin, C. (2012). Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results. In *2012 IEEE International Conference on Robotics and Automation*, 279–284.

Chen, S., Saulnier, K., Atanasov, N., Lee, D.D., Kumar, V., Pappas, G.J., and Morari, M. (2018). Approximating explicit model predictive control using constrained neural networks. In *2018 Annual American Control Conference (ACC)*, 1520–1527. IEEE.

De Schutter, B. and De Moor, B. (1998). Optimal traffic light control for a single intersection. *European Journal of Control*, 4(3), 260–276.

Erez, T., Lowrey, K., Tassa, Y., Kumar, V., Kolev, S., and Todorov, E. (2013). An integrated system for real-time model predictive control of humanoid robots. In *2013 13th IEEE-RAS International conference on humanoid robots (Humanoids)*, 292–299. IEEE.

Geyer, T., Torrisi, F.D., and Morari, M. (2008). Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7), 1728–1740.

Gilbert, E.G., Kolmanovsky, I., and Tan, K.T. (1994). Nonlinear control of discrete-time linear systems with state and control constraints: A reference governor with global convergence properties. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, 144–149.

Herceg, M., Kvasnica, M., Jones, C.N., and Morari, M. (2013). Multi-parametric toolbox 3.0. In *2013 European Control Conference (ECC)*, 502–510.

Holaza, J., Klaučo, M., Drgoňa, J., Oravec, J., Kvasnica, M., and Fikar, M. (2018). MPC-based reference governor control of a continuous stirred-tank reactor. *Computers & Chemical Engineering*, 108, 289–299.

Klaučo, M., Kaluz, M., and Kvasnica, M. (2017). Real-time implementation of an explicit MPC-based reference governor for control of a magnetic levitation system. *Control Engineering Practice*, 60, 99–105.

Klaučo, M. and Kvasnica, M. (2017). Control of a boiler-turbine unit using MPC-based reference governors. *Applied Thermal Engineering*, 110, 1437–1447.

Limón, D., Alamo, T., de la Peña, D.M., Zeilinger, M.N., Jones, C., and Pereira, M. (2012). MPC for tracking periodic reference signals. *IFAC Proceedings Volumes*, 45(17), 490–495.

Limón, D., Alvarado, I., Alamo, T., and Camacho, E.F. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9), 2382–2387.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986.

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

Morari, M. and Lee, J.H. (1999). Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5), 667–682.

Pistikopoulos, E.N., Dua, V., Bozinis, N.A., Bemporad, A., and Morari, M. (2002). On-line optimization via off-line parametric optimization tools. *Computers & Chemical Engineering*, 26(2), 175–185.

Qin, S.J. and Badgwell, T.A. (2000). An overview of nonlinear model predictive control applications. *Nonlinear Model Predictive Control*, 369–392.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.

Richter, C., Vega-Brown, W., and Roy, N. (2018). Bayesian learning for safe high-speed navigation in unknown environments. In *Robotics Research*, 325–341. Springer.

Sadowska, A., van Overloop, P.J., Maestre, J.M., and De Schutter, B. (2015). Human-in-the-loop control of an irrigation canal using time instant optimization model predictive control. In *2015 European Control Conference (ECC)*, 3274–3279.

Tøndel, P. and Johansen, T.A. (2002). Complexity reduction in explicit linear model predictive control. *IFAC Proceedings Volumes*, 35(1), 189–194.

van Ekeren, H., Negenborn, R., van Overloop, P., and De Schutter, B. (2013). Time-instant optimization for hybrid model predictive control of the rhine–meuse delta. *Journal of Hydroinformatics*, 15(2), 271–292.

Watterson, M. and Kumar, V. (2015). Safe receding horizon control for aggressive MAV flight with limited range sensing. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3235–3240.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4(2), 65–85.