J. Appl. Numer. Optim. 5 (2023), No. 2, pp. 201-217
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.5.2023.2.02

MODEL PREDICTIVE CONTROL OF DISCRETE-TIME LINEAR SYSTEMS BY ADMM WITH APPLICATIONS TO TURBOFAN ENGINE CONTROL PROBLEMS

JIAO TENG[1], XIAN DU[2], LEI WANG[3,*], XINWEI WANG[4], JUNMEI WU[3]

[1]*Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong*
[2]*School of Control Science and Engineering, Dalian University of Technology, Dalian, China*
[3]*School of Mathematical Sciences, Dalian University of Technology, Dalian, China*
[4]*Department of Engineering Mechanics, State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian, China*

Abstract. In this paper, we consider optimal control problems with linear discrete state space model, which originate from a class of turbofan engines. The optimization problem associated with each moving horizon estimation (MHE) in classical model predictive control (MPC) is a quadratic programming (QP) problem, and the general QP algorithms does not exploit the structural features of the turbofan engine itself to improve the computational efficiency of the algorithm. In the framework of model predictive control, the turbofan engine model makes the rolling optimization subproblem exhibit a sparse structure. Based on this feature, the alternating direction method of multipliers (ADMM) is employed to solve each optimization sub-problem and design an improved MPC-ADMM algorithm for solving this class of optimal control problems. The simulation results are compared with the MPC-QP algorithm by numerical examples to show the effectiveness and superiority of the novel algorithm.

Keywords. Model predictive control; MPC-ADMM algorithm; Optimal control problems; Turbofan engine control problems.

2020 Mathematics Subject Classification. 90C20, 49N05.

*Corresponding author.
E-mail addresses: jtengmath@163.com (J. Teng), duxian@dlut.edu.cn (X. Du), wanglei@dlut.edu.cn (L. Wang),wangxinwei@dlut.edu.cn (X.W. Wang), 15738519912@163.com (J. Wu).

Received March 8, 2022; Accepted January 14, 2023.

1. Introduction

The aircraft engine is the core component of aircrafts. The safe, reasonable, effective, and stable operation and best performance of the engine cannot be achieved without reliable control technology. Therefore, it is significant and essential to improve the control technology of the control system for aero-engine. Conventional linear regulators are unable to the control systems with complex output limit protection as a result of their conservative nature; see, e.g., [1]. Model Predictive Control (MPC) which first appeared in the late 1970s and is now known as rolling horizon control [2], is a type of control algorithm with excellent dynamic control performance [3]. In recent years, MPC algorithms received wide attention in the field of engine control [4, 5] because of its advantages of explicitly processing engine constraints [6, 7], simplifying the engine control system structure [8], and achieving real-time rolling optimization. Industrial control processes are often accompanied by nonlinearities and uncertainties, and MPC not only indicates theoretical ability to handle the relevant constraints, but also has strong practicality. Indeed, MPC has been widely used in numerous fields; see, e.g., [9, 10].

MPC essentially requires solving a series of optimization problems [11, 12]. Different performance index in real-world problems generally require different optimization algorithms. Traditionally, 1-norm and infinite-norm performance functions use linear programming optimization algorithms; quadratic performance metrics, which in the most common case use quadratic programming algorithms. Commonly used quadratic programming algorithms in MPC are the interior point method [13, 14] and the active-set method [15]. Although MPC is able to consider the constraints in systems, the corresponding optimization subproblems need to be solved online in real time to obtain their optimal control sequences at each sampling time, which will increase the computational and storage capacity and limit the application of MPC in large-scale practical problems. Therefore, computational efficiency is crucial for solving MPC problems. There have been many improved works in this area, for example, the extended Newton Raphson algorithm [16], the gradient algorithm [17], and the ADMM algorithm (the common MPC problems: the Lasso MPC problem for time-varying systems [18], the MPCT problem [19], the MPC problem for systems with feedback gain [20], the BCMPC problem [21], and the symmetric MPC problem [22]).

The traditional approach to solve the turbofan engine MPC problem usually converts it into a series of quadratic programming problems, then solves with the assistance of the SNOPT based on active set methods and the IPOPT based on interior methods, however, the dimensionality of the variables of the quadratic programming problem depends on the choice of the prediction and control horizon, which will bring two dilemmas: 1) it is difficult to obtain its analytical solution directly for the quadratic programming problem with inequality constraints; 2) as the control and prediction horizon keep increasing, the number of inequality constraints increases, and at each sampling, the quadratic programming subproblem needs to repeatedly solve the corresponding KKT system, which will greatly reduce the problem solving efficiency.

In this paper, an improved MPC-ADMM algorithm is obtained from the structural characteristics of the turbofan aero-engine itself based on the idea of ADMM algorithms, which make full use of the distributed optimization characteristics of ADMM and the sparsity of MPC problems to improve the efficiency of solving such optimal control problems. ADMM was first proposed by Glowiski and Marrocco [23], and Gabay and Mercier [24] in the mid-1970s. The basic idea of the ADMM algorithm is to reduce the difficulty and cost of solving a large-scale 2-block problem by transforming it into a number of small-scale problems. Its primary advantages are (1) in theory, convergence is guaranteed for any convex-valued function and constraints [25, 26] and (2) in practice, the enhanced Lagrangian term usually accelerates the convergence rate. The first-order properties of the ADMM algorithm, its separable structure, and its potential distributive computing power can be of great benefit to the solution and the study of large-scale problems.

Recently, there are some new results on ADMM algorithms for MPC problems. However, there are still some open problems in these studies that have not been solved yet. In [27], the objective function is a conventional finite-time domain quadratic function. In [28, 29], the output variables and the corresponding output restrictions are not considered. In [30], the output of the system does not consider the control input or the direct effect of the control input on the output. In [31], the bound constraints on the system outputs and control inputs are not taken into account. In addition, there are few literatures on the applications of MPC-ADMM in the context of practical aero-engines. Therefore, in contrast to the previous existing works, we consider the optimal control problem for a linear discrete state space model of a turbofan aero-engine, including the output variables and the corresponding output constraints, while considering the bound constraints on the control inputs and the system outputs. The simulation results of the improved MPC-ADMM algorithm and the MPC-QP algorithm are compared through simulation experiments with numerical examples, which illustrate the effectiveness of the algorithm.

The remainder of this paper is organized as follows. In Section 2, we describe the linear discrete state-space model of the turbofan engine. In Section 3, we perform state and output predictions and transform the proposed optimal control problem into a linear programming problem with inequality constraints. In Section 4, we apply the ADMM algorithm to MPC rolling optimization, present the MPC-ADMM algorithm, and provide the detailed steps. In Section 5, we present and analyze the comparison results to show the efficiency of our method. To make the article more brief, we put some of the complex derivations of the equations in the appendix at the end. Section 6 ends this paper.

2. TURBOFAN ENGINE MODEL DESCRIPTION

The linear discrete state space model for turbofan engines are as follows

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k), \\ y(k) &= C_d x(k) + D_d u(k), \end{aligned} \tag{2.1}$$

where

$$x = \begin{bmatrix} \Delta N_f & \Delta N_c \end{bmatrix}^T, \; y = \begin{bmatrix} \Delta T_{48} & \Delta \text{SmHPC} \end{bmatrix}^T, \; u = \begin{bmatrix} \Delta W_f & \Delta \text{VSV} & \Delta \text{VBV} \end{bmatrix}^T,$$

N_f denotes the angular speed of the assembly formed by fan, LPC, and LPT, and N_c denotes the angular speed of the HPC-HPT shaft assembly. For the consistency with the standard terminology, N_f and N_c denotes fan speed and core speed, respectively. Outlet temperature of high pressure compressor T_{48} and high pressure compressor stall margin ΔSmHPC are regarded as outputs. The control variables are the deviation of fuel flow W_f from the steady state, the deviation of the adjustable stator blade Angle VSV, and the deviation of adjustable opening of the vent valve VBV, respectively.

3. PROBLEM REFORMULATION BASED ON MODEL PREDICTIVE CONTROL

3.1. **Model-based prediction equations.** To eliminate or reduce static error, we rewrite model (2.1) as an incremental model

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} &= \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} B_d \\ I \end{bmatrix} \Delta u(k), \\ y(k) &= \begin{bmatrix} C_d & D_d \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + D_d \Delta u(k), \end{aligned} \tag{3.1}$$

where

$$\Delta u(k) = u(k) - u(k-1).$$

Denoting the augmented state vector as $x_a(k) = \left[x(k)^T \; u(k-1)^T \right]^T$, the compact form of model (3.1) is

$$\begin{aligned} x_a(k+1) &= A_{da} x_a(k) + B_{da} \Delta u(k), \\ y(k) &= C_{da} x_a(k) + D_{da} \Delta u(k), \end{aligned} \tag{3.2}$$

where

$$A_{da} = \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix}, \; B_{da} = \begin{bmatrix} B_d \\ I \end{bmatrix}, \; C_{da} = \begin{bmatrix} C_d & D_d \end{bmatrix}, \; D_{da} = D_d.$$

Assumption 3.1. Since the prediction of the future dynamics of the system requires control inputs for the entire prediction horizon, the following assumptions are added

- n_u is the control horizon, n_y is the prediction horizon, and satisfies $n_u \le n_y$;

- $\Delta u(k+j) = 0, j = n_u, n_u+1, \cdots, n_y$, that is, outside the control time domain, the control variable is 0.

Inspired by [32], the predicted states and the predicted output column can be obtained by iterating model (3.2). Then the output of future prediction of the system can be calculated by the following prediction equation:

$$\begin{aligned} \hat{x}_a &= P_x x_a(k) + H_x \Delta\hat{u}, \\ \hat{y} &= P x_a(k) + H \Delta\hat{u}. \end{aligned} \tag{3.3}$$

The derivation of the predicted states and predicted output column and the specific matrix of coefficients can be found in the appendix.

3.2. **Finite-time domain optimization problem with time forward rolling.** The choice of the objective function reflects the requirements for the system performance. In the optimal control problem of turbofan engine, we want the system to be highly traceable, i.e., we want to find the optimal control so that the output of the predicted system is sufficiently close to the output of the expected system. In addition, we also want the fluctuation of the control to be as narrow as possible, so we add a quadratic form of the control variation to the objective function, as the form below,

$$J = \sum_{i=1}^{n_y} e(k+i)^T e(k+i) + \sum_{i=0}^{n_u-1} a\Delta u(k+i)^T \Delta u(k+i), \tag{3.4}$$

where $e(k) = r(k) - \hat{y}(k)$, $r(k)$ is the vector of reference inputs (the given fan speed deviation value), and a is the control weighting factor.

Assumption 3.2. In the process of tracking reference inputs, the control system needs satisfy the following conditions,

- The deviation of fuel flow W_f, the deviation of adjustable stator blade Angle ΔVSV, and the deviation of adjustable opening of the vent valve ΔVBV are all within the acceptable limits.
- The deviation of the outlet temperature of a high pressure compressor T_{48} and the deviation of surge margin of high pressure compressor $\Delta SmHPC$ are all remain within the allowable limits.

that is,

$$\begin{aligned} &\underline{U} \le u(k+i) \le \overline{U}, i = 0,1,2,\cdots,n_u-1, \\ &\underline{Y} \le y(k+i) \le \overline{Y}, i = 1,2,\cdots,n_y, \end{aligned} \tag{3.5}$$

which $\underline{U}$, $\overline{U}$, $\underline{Y}$, and $\overline{Y}$ represent vectors containing boundary values of control quantity and output quantity, respectively.

At each sampling time, the control needs to be obtained by solving an optimization problem. Differing from the discrete optimal control algorithm, the predictive control does not adopt a constant global optimization objective, but a finite-time domain optimization strategy with time rolling forward. At each sampling time k, substitute $\hat{y}(k) = Px_a(k) + H\Delta\hat{u}$ into objective function (3.4), and combine with constraints (3.5). The rolling optimization problems need to be solved are as follows.

Problem. ROP1

$$\begin{array}{ll} \min_{\Delta\hat{u}} & J = (r - Px_a(k) - H\Delta\hat{u})^T (r - Px_a(k) - H\Delta\hat{u}) + a(\Delta\hat{u})^T \Delta\hat{u} \\ \text{s.t} & \underline{U} \leq u(k+i) \leq \overline{U}, i = 0, 1, 2, \cdots, n_u - 1, \\ & \underline{Y} \leq y(k+i) \leq \overline{Y}, i = 1, 2, \cdots, n_y. \end{array}$$

Next, we express the input and output constraints as the functions of the form with respect to $\Delta\hat{u}$. We transform the objective function and constraints of Problem **ROP1** as follows.

For the objective function, it is directly simplified to obtain:

$$J = \Delta\hat{u}^T \left(H^T H + aI\right) \Delta\hat{u} + 2\left(x_a(k)^T P^T H - r^T H\right) \Delta\hat{u} + J_0, \tag{3.6}$$

where $J_0 = r^T r - 2r^T Px_a(k) + x_a(k)^T P^T Px_a(k)$. The matrix form of control constraint conditions are:

$$\begin{aligned} C_c \Delta\hat{u} &\leq \overline{d_u}, \\ -C_c \Delta\hat{u} &\leq \underline{d_u}. \end{aligned} \tag{3.7}$$

The matrix form of control constraint conditions are:

$$M\Delta\hat{u} \leq d. \tag{3.8}$$

The detailed derivation of Equations (3.7) and (3.8) can be found in Appendix Part II.

Combining the above equation with cost function (3.6), we can obtain the quadratic programming problem with inequality constraint.

Problem. ROP2

$$\begin{array}{ll} \min\limits_{\Delta\hat{u}} & J = \Delta\hat{u}^T \left(H^T H + aI\right) \Delta\hat{u} + 2\left(x_a(k)^T P^T H - r^T H\right) \Delta\hat{u} \\ \text{s.t} & M\Delta\hat{u} \leq d. \end{array}$$

Problem **ROP2** is a quadratic programming problem with inequality constraints. The quadprog function in Matlab (Optimization Toolbox) can be used to handle such optimization problems. However, in practice, the direct use of Matlab leads to a significant reduction in computational efficiency as the control and prediction time domains are extended and the number of inequality constraints increases. Therefore, in order to improve computational efficiency, we consider applying the ADMM algorithm to the MPC rolling optimization problem based on the specificity of the structure of the aero-engine model in the following section.

4. Numerical Solutions via MPC-ADMM Algorithm

4.1. **ADMM algorithm applied to the quadratic programming problem ROP2.** First, we transform the quadratic programming problem **ROP2** with inequality constraints into equality constraints and then apply the ADMM algorithm. We also give the convergence analysis. By adding the relaxation variable $z(\geq 0)$ to the inequality constraints, Problem **ROP2** can be transformed into a quadratic programming problem with the equality constraints.

Problem. ROP3

$$\begin{aligned} &\min_{\Delta\hat{u}} J = \Delta\hat{u}^T \left(H^T H + aI\right) \Delta\hat{u} + 2\left(x_a(k)^T P^T H - r^T H\right)\Delta\hat{u} \\ &\text{s.t} \quad M\Delta\hat{u} + z = d, (z \geq 0). \end{aligned} \tag{4.1}$$

Let the current iteration point be $(\Delta\hat{u}^s, z^s, \lambda^s)$. By the ADMM algorithm framework [33], the following iterative formula can be obtained

$$\begin{cases} \Delta\hat{u}^{s+1} = \arg\min\left\{L_\beta\left(\Delta\hat{u}, z^s, \lambda^s\right)\right\}, \\ z^{s+1} = \arg\min\left\{L_\beta\left(\Delta\hat{u}^{s+1}, z, \lambda^s\right) \mid z \geq 0\right\}, \\ \lambda^{s+1} = \lambda^s + \beta\left(M\Delta\hat{u}^{s+1} + z^{s+1} - d\right), \end{cases} \tag{4.2}$$

where L_β is the augmented Lagrangian function

$$\begin{aligned} L_\beta(\Delta\hat{u}, z, \lambda) = {} & \Delta\hat{u}^T \left(H^T H + aI\right)\Delta\hat{u} + 2\left(x_a(k)^T P^T H - r^T H\right)\Delta\hat{u} \\ & + \lambda^T (M\Delta\hat{u} + z - d) + \frac{\beta}{2}\|M\Delta\hat{u} + z - d\|^2. \end{aligned}$$

By simplifying iterative formula (4.2), we arrive at

$$\begin{cases} \Delta\hat{u}^{s+1} = -\left[2H^T H + 2aI + \beta M^T M\right]^{-1}\left[2H^T\left(Px_a - r\right) + M^T\left(\lambda^s + \beta\left(z^s - d\right)\right)\right], \\ z^{s+1} = \max\left\{0, -\frac{\lambda^s + \beta\left(M\Delta\hat{u}^{s+1} - d\right)}{\beta}\right\}, \\ \lambda^{s+1} = \lambda^s + \beta\left(M\Delta\hat{u}^{s+1} + z^{s+1} - d\right). \end{cases} \tag{4.3}$$

The optimal solution $\Delta\hat{u}$ can be obtained by cyclic iterations based on (4.3). Based on the system model and the first component of $\Delta\hat{u}$, we can calculate the state and output quantities at $k+1$ moments, which can be used as the initial condition to solve the optimal control sequence at $k+1$ moments, and the cycle iterates until the end of the whole simulation time domain. Following the Section 3 of [33], it is not difficult to prove the following convergence theorem.

Theorem 4.1. *$J\left(\Delta\hat{u}^k\right) \to J^\star$ as $k \to \infty$, i.e., the objective function of the iterates approaches the optimal value.*

Theorem 4.1 guarantees the convergence of the ADMM algorithm. It is worth noting that since the objective function that we consider is strictly convex, the convergence of the optimal solution is also guaranteed. Next, we apply the ADMM algorithm to the rolling optimization process of MPC, propose the MPC-ADMM algorithm, and provide the detailed steps.

4.2. **MPC-ADMM algorithm design.** Based on the previous discussion, we propose an improved MPC algorithm, MPC-ADMM algorithm, under the MPC problem corresponding to linear discrete state-space models of turbofan engine. In order to define the distance of each iteration point to the optimality system of the original problem, the primal residual and the dual residual are defined in ADMM iteration. Similar to the derivation in [33], we can obtain the reasonable termination criterion is that the primal residual $\left\|M\Delta\hat{u}^{s+1}+z^{s+1}-d\right\|_2$ and dual residual $\left\|\beta M^T\left(z^{s+1}-z^s\right)\right\|_2$ must be small. The algorithm are given as follows

---

Algorithm 1 MPC-ADMM algorithm

Input: Constant matrix A_d, B_d, C_d, D_d; upper and lower bounds $\overline{U}, \underline{U}$, $\overline{Y}, \underline{Y}$; control time domain n_u, prediction time domain n_y; simulation time domain *simhor*, $\varepsilon^{pri}, \varepsilon^{dual}, \beta > 0, s = 0, a, r$.

Output: Ensemble of classifiers on the current batch, E_n;

1: Given $x_a(k)$ and $x(k)$, initial u.
2: Compute $P, H, C_c, L, M, y, \overline{d_u}, \underline{d_u}, \overline{d_y}, \underline{d_y}, d$.
3: Given $(\Delta\hat{u}^s, z^s, \lambda^s)$, obtain $\left(\Delta\hat{u}^{s+1}, z^{s+1}, \lambda^{s+1}\right)$
4: **if** $\left\|M\Delta\hat{u}^{s+1}+z^{s+1}-d\right\|_2 \geq \varepsilon^{pri}$ and $\left\|\beta M^T\left(z^{s+1}-z^s\right)\right\|_2 \geq \varepsilon^{dual}$, **then**
5: $\quad z^s = z^{s+1}$, $\lambda^s = \lambda^{s+1}$
6: **else** Take the optimal control sequence as $\Delta\hat{u}^{s+1}$
7: **end if**
8: Take the first component $\Delta u(k)$ of $\Delta\hat{u}(k)$, and obtain the current time optimal control $u(k)$;
9: Calculate state quantity $x(k+1)$ and output quantity $y(k+1)$;
10: Calculate augmented state $x_a(k+1)$, and return to Step 2 until the end of *simhor* in the whole simulation time domain.

---

5. Numerical Results

The example is taken from [6], which uses the CMAPSS-40K nonlinear engine model near ground idle. The optimization objective is to generate a fast fan speed response $\Delta N_f = 100\text{r/min}$. At the same time, the control quantities $\Delta W_f, \Delta\text{VSV}, \Delta\text{VBV}$ are kept within acceptable limits. In addition, both output quantities $\Delta T_{48}, \Delta\text{SmHPC}$ need to be kept within the allowable range.

In this example, the parameters are set as follows: the control domains $n_u = 3$, the prediction time domains $n_y = 7$, the simulation time $simhor = 30$, the control weighting factor $a = 0.01$, the upper and lower bounds of control and output are $\underline{U} = [-1; -20; -0.5]$, $\overline{U} = [2; 15; 0.4]$, $\underline{Y} = [-150; -10]$, $\overline{Y} = [300; 20]$, $\varepsilon^{pri} = 0.07$, $\varepsilon^{dual} = 0.01$,

$$A_d = \begin{pmatrix} -3.3808 & 1.2954 \\ 0.4444 & -3.0501 \end{pmatrix}, \qquad B_d = \begin{pmatrix} 667.8408 & -39.2134 & -14.2485 \\ 1334 & 117.2730 & -26.8107 \end{pmatrix},$$
$$C_d = \begin{pmatrix} -0.0191 & -0.1178 \\ 0.0158 & -0.0037 \end{pmatrix}, \qquad D_d = \begin{pmatrix} 289.0525 & 0.1332 & 1.2568 \\ -10.9483 & 0.8137 & -0.4766 \end{pmatrix}.$$

It is worth noting that the parameters in this example are consistent with those in [6] for comparison purposes.

5.1. Numerical results of the MPC-ADMM algorithm. The results of the control are demonstrated in Fig. 1, and the control value does not exceed its given upper and lower limits in the process of reaching the target value of the wind speed response. It can be seen from Fig. 2 that the tracking quantity ΔN_f reaches the target response value of 100r/min in 0.06s with the two outputs ΔT_{48} and SMHPC, which do not exceed the allowed limit values.

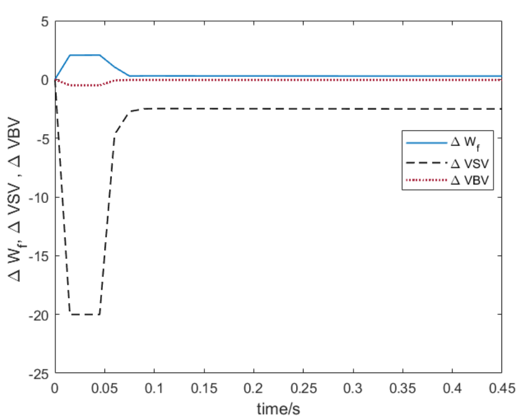

FIGURE 1. Control inputs

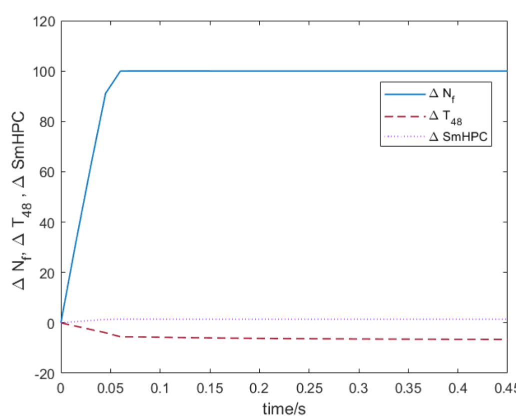

FIGURE 2. Output response

5.2. **Comparison between MPC-ADMM algorithm and MPC-QP algorithm.** In order to demonstrate the superiority of MPC-ADMM algorithm more directly, we compare the results of in [6] with the simulation results of this paper in detail. In the listed figures and tables, the results of MPC-QP are labeled with QP, the results of MPC-ADMM are labeled with ADMM.

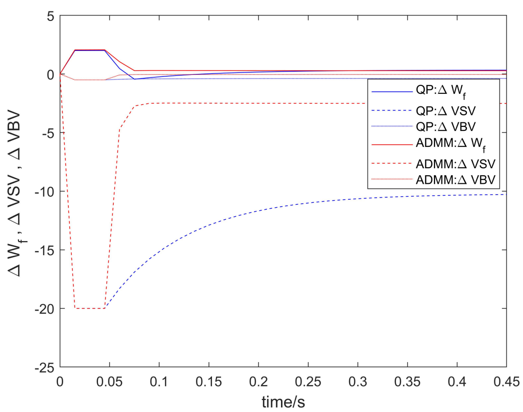

FIGURE 3. Comparison on control variables

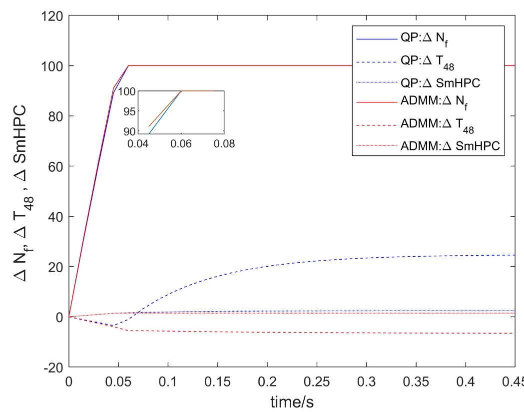

FIGURE 4. Comparison on output variables

Both the MPC-ADMM and the MPC-QP algorithms obtain control that are within their corresponding control ranges when tracking target values. Here we mainly compare the differences between the two algorithms. As demonstrated in Fig. 3 and Fig. 4, in terms of the control quantity ΔW_f, the control curves of the MPC-ADMM algorithm are lower than those of the MPC-QP algorithm within 0.04s and 0.08s, and the tracking value ΔN_f of the MPC-ADMM algorithm is higher than that of the MPC-QP algorithm. In other words, under the framework of MPC for turbofan aero-engines, the MPC-ADMM algorithm obtains a faster response in terms of tracking quantities within 0.04s and 0.08s. On the other hand, we gradually increase the prediction horizon n_y and control horizon n_u under the condition that the control constraint and output constraint are satisfied, and perform numerical experiments separately to obtain a comparative table of CPU consumption time (Table 1) and the corresponding graph (Fig. 5) for the two algorithms. For the convenience of discussion, we treat n_y and n_u as a whole and number them as serial number 1, serial number 2.... As the numbering increases, the corresponding time domain gradually increases, and the CPU consumption time of the two algorithms increases. Note that the dimensions of the decision variables and constraints demonstrate a positive correlation with the size of the control and prediction time domains, so that when the time domain increases, the scale of the optimization problem to be solved becomes larger. The MPC-ADMM algorithm consumes less CPU time than the MPC-QP algorithm, and the larger the time domain, the bigger the difference. Therefore, these results reflect the advantages of the MPC-ADMM algorithm in terms of less time consuming and faster response relative to the MPC-QP algorithm, which further demonstrates the effectiveness and superiority of the MPC-ADMM algorithm.

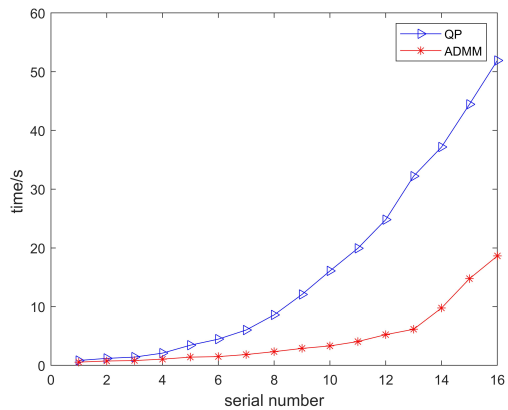

FIGURE 5. Relationship of serial number and time consumption

6. CONCLUSION

In this paper, the ADMM algorithm was embedded in the framework of the MPC to solve the optimal control problem with turbofan engines. An improved MPC-ADMM algorithm was proposed by transforming the large scale problem into a series of small scale optimization subproblems by using the idea of the alternate projection method. Numerical results demonstrate that the improved MPC-ADMM algorithm can not only make the tracking volume have higher response speed, but also reduce the CPU running time of the system compared with the classical algorithm of quadratic programming, which greatly improves the computational efficiency.

Acknowledgments
The author is grateful to the referee for useful suggestions which improved the contents of this paper. This work was supported by the National Natural Science Foundation of China under Grant No. 12161076, 61890920, 61890921, and 12102077 and the Fundamental Research Funds for the Central Universities No.DUT21LAB301 and DUT22LAB305.

TABLE 1. Comparison table of CPU(s) time consumption in different horizons

Number	Prediction time dom.	Control time dom.	ADMM/CPU(s)	QP/CPU(s)
1	$n_y = 7$	$n_u = 3$	0.57920	0.83940
2	$n_y = 10$	$n_u = 6$	0.76740	1.19160
3	$n_y = 14$	$n_u = 10$	1.08670	1.41430
4	$n_y = 18$	$n_u = 14$	1.48750	2.08630
5	$n_y = 22$	$n_u = 18$	2.47020	3.43940
6	$n_y = 26$	$n_u = 22$	2.61850	4.46470
7	$n_y = 30$	$n_u = 26$	3.26190	6.02600
8	$n_y = 34$	$n_u = 30$	4.12608	8.59700
9	$n_y = 38$	$n_u = 34$	5.56626	12.07720
10	$n_y = 42$	$n_u = 38$	6.78416	16.08448
11	$n_y = 46$	$n_u = 4$	8.20898	19.93004
12	$n_y = 50$	$n_u = 46$	10.06792	24.80638
13	$n_y = 54$	$n_u = 50$	12.18096	32.22338
14	$n_y = 58$	$n_u = 54$	14.32200	37.17458
15	$n_y = 62$	$n_u = 58$	17.18480	44.45082
16	$n_y = 66$	$n_u = 62$	21.20732	51.91660

REFERENCES

[1] X. Du, H. Richter, Y. Guo, Multivariable sliding-mode strategy with output constraints for aeroengine propulsion control, Journal of Guidance, Control, and Dynamics, 39 (2016), 1631-1642.

[2] J. Richalet, A. Rault, J.L. Testud, et al., Model predictive heuristic control: Applications to industrial processes, Automatica 14 (1978), 413-428.

[3] J.H. Lee, Model predictive control: Review of the three decades of development, Int. J. Control, Auto. Sys. 9 (2011), 415-424.

[4] X. Du, X.M. Sun, Z.M. Wang, et al., A scheduling scheme of linear model predictive controllers for turbofan engines, IEEE Access 5 (2017), 24533-24541.

[5] J. Seok, I. Kolmanovsky, A. Girard, Coordinated model predictive control of aircraft gas turbine engine and power system, J. Guidance, Control, Dynamics, 40 (2017), 2538-2555.

[6] H. Richter, Advanced Control of Turbofan Engines, Springer, New York, 2012.

[7] J. Mu, D. Rees, G.P. Liu, Advanced controller design for aircraft gas turbine engines, Control Engineering Practice, 13 (2005), 1001-1015.

[8] A. Aly, I. Atia, Neural modeling and predictive control of a small turbojet engine (SR-30), In: 10th International Energy Conversion Engineering Conference, Atlantia, 2012.

[9] S.M. Lee, H. Kim, H. Myung, et al., Cooperative coevolutionary algorithm-based model predictive control guaranteeing stability of multirobot formation, IEEE Trans. Control Systems Technology 23 (2014), 37-51.

[10] P.M. Marusak, Numerically efficient fuzzy MPC algorithm with advanced generation of prediction—application to a chemical reactor, Algorithms 13 (2020), 143.

[11] D. Krishnamoorthy, E. Suwartadi, B. Foss, et al., Improving scenario decomposition for multistage MPC using a sensitivity-based path-following algorithm, IEEE Control Systems lett. 2 (2018), 581-586.

[12] I. Necoara, D. Clipici, Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC, J. Process Control 23 (2013), 243-253.

[13] Y. Wang, S. Boyd, Fast model predictive control using online optimization, IEEE Trans. Control Systems Techno. 18 (2009), 267-278.

[14] M.S.K. Lau, S.P. Yue, K.V. Ling, et al., A comparison of interior point and active set methods for FPGA implementation of model predictive control, 2009 European Control Conference (ECC), pp. 156-161, IEEE, 2009.

[15] A.G. Wills, G. Knagge, B. Ninness, Fast linear model predictive control via custom integrated circuit architecture, IEEE Trans. Control Systems Techno. 20 (2011), 59-71.

[16] B. Kouvaritakis, M. Cannon, J.A. Rossiter, Who needs QP for linear MPC anyway? Automatica, 38 (2002), 879-884.

[17] Y. Yu, P. Elango, B. Açıkmeşe, Proportional-integral projected gradient method for model predictive control, IEEE Control Systems Lett. 5 (2020), 2174-2179.

[18] Z. Cheng, J. Ma, X. Zhang, et al., Semi-proximal admm for model predictive control problem with application to a UAV system[C]//2020 20th International Conference on Control, Automation and Systems (ICCAS), pp. 82-87, IEEE, 2020.

[19] P Krupa, I Alvarado, D Limon, et al, Implementation of model predictive control for tracking in embedded systems using a sparse extended ADMM algorithm[J]. IEEE Transactions on Control Systems Technology, 2021.

[20] F. Lin, M. Fardad, M.R. Jovanović, Sparse feedback synthesis via the alternating direction method of multipliers, In: 2012 American Control Conference (ACC), pp. 4765-4770, IEEE, 2012.

[21] I. Kempf, P.J. Goulart, S. Duncan, Alternating direction of multipliers method for block circulant model predictive control, In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 4311-4316, IEEE, 2019.

[22] C Danielson, An alternating direction method of multipliers algorithm for symmetric model predictive control[J]. Optimal Control Applications and Methods, 2021, 42(1): 236-260.

[23] R. Glowinski, A. Marroco, Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique 9 (1975), 41-76.

[24] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl. 2 (1976), 17-40.

[25] R. Nishihara, L. Lessard, B. Recht, et al., A general analysis of the convergence of ADMM, In: International Conference on Machine Learning, pp. 343-352, PMLR, 2015.

[26] W. Shi, Q. Ling, K. Yuan, et al., On the linear convergence of the ADMM in decentralized consensus optimization, IEEE Trans. Signal Process. 62 (2014), 1750-1761.

[27] Y. Li , Q. Zou, X. Ji, et al., Fast model predictive control based on adaptive alternating direction method of multipliers, J. Chemistry, 2019 (2019), 8035204.

[28] S East, M Cannon, ADMM for MPC with state and input constraints, and input nonlinearity[C]//2018 annual American control conference (ACC). IEEE, 2018: 4514-4519.

[29] M. Schulze Darup, G. Book, On Closed-Loop Dynamics of ADMM-Based MPC, In: Faulwasser, T., Müller, M.A., Worthmann, K. (eds) Recent Advances in Model Predictive Control. Lecture Notes in Control and Information Sciences, vol 485. Springer, 2021.

[30] Y. Guo, H. Gao, H. Xing, et al., Decentralized coordinated voltage control for VSC-HVDC connected wind farms based on ADMM, IEEE Trans. Sustainable Energy 10 (2018), 800-810.

[31] S. Huang, Q. Wu, W. Bao, et al., Hierarchical optimal control for synthetic inertial response of wind farm based on alternating direction method of multipliers, IEEE Trans. Sustainable Energy 12 (2020), 25-35.

[32] E.F. Camacho, C.B. Alba, Model Predictive Control, Springer, London, 2013.

[33] S. Boyd, N. Parikh, E. Chu, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine learning, 3 (2011), 1-122.

APPENDIX

Part I Derivation of Equation (3.3). The predicted states are as follows:

$$\begin{aligned}
x_a(k+1) &= A_{da}x_a(k) + B_{da}\Delta u(k),\\
x_a(k+2) &= A_{da}x_a(k+1) + B_{da}\Delta u(k+1)\\
&= A_{da}^2 x_a(k) + A_{da}B_{da}\Delta u(k) + B_{da}\Delta u(k+1),\\
&\vdots\\
x_a(k+n_u) &= A_{da}x_a(k+n_u-1) + B_{da}\Delta u(k+n_u-1)\\
&= A_{da}^{n_u}x_a(k) + A_{da}^{n_u-1}B_{da}\Delta u(k) + A_{da}^{n_u-2}B_{da}\Delta u(k+1)\\
&+\cdots + B_{da}\Delta u(k+n_u-1),\\
&\vdots\\
x_a(k+n_y) &= A_{da}x_a(k+n_y-1) + B_{da}\Delta u(k+n_y-1)\\
&= A_{da}^{n_y}x_a(k) + A_{da}^{n_y-1}B_{da}\Delta u(k) + A_{da}^{n_y-2}B_{da}\Delta u(k+1)\\
&+\cdots + A_{da}^{n_y-n_\alpha}B_{da}\Delta u(k+n_u-1).
\end{aligned}$$

The predicted output column can be obtained:

$$\begin{aligned}
y(k+1) =& C_{da}x_a(k+1) + D_{da}\Delta u(k+1)\\
=& C_{da}A_{da}x_a(k) + C_{da}B_{da}\Delta u(k) + D_{da}\Delta u(k+1),\\
y(k+2) =& C_{da}x_a(k+2) + D_{da}\Delta u(k+2) + C_{da}A_{da}^2 x_a(k)\\
=& C_{da}A_{da}B_{da}\Delta u(k) + C_{da}B_{da}\Delta u(k+1) + D_{da}\Delta u(k+2),\\
&\vdots\\
y(k+n_u) =& C_{da}x_a(k+n_u) + D_{da}\Delta u(k+n_u) + C_{da}A_{da}^{n_u}x_a(k)\\
=& C_{da}A_{da}^{n_u-1}B_{da}\Delta u(k) + C_{da}A_{da}^{n_z-2}B_{da}\Delta u(k+1)\\
&+\cdots + C_{da}B_{da}\Delta u(k+n_u-1),\\
&\vdots\\
y(k+n_y) =& C_{da}x_a(k+n_y) + D_{da}\Delta u(k+n_y) + C_{da}A_{da}^{n_y}x_a(k)\\
=& C_{da}A_{da}^{n_y-1}B_{da}\Delta u(k) + C_{da}A_{da}^{n_y-2}B_{da}\Delta u(k+1)\\
&+\cdots + C_{da}A_{da}^{n_y-n_u}B_{da}\Delta u(k+n_u-1).
\end{aligned}$$

The output of future prediction of the system can be calculated by the following prediction equation:

$$\hat{x}_a = P_x x_a(k) + H_x \Delta\hat{u},$$
$$\hat{y} = P x_a(k) + H \Delta\hat{u},$$

where

$$\hat{x}_a = \begin{bmatrix} x_a(k+1) \\ \vdots \\ x_a(k+n_y) \end{bmatrix}, \hat{y} = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+n_y) \end{bmatrix},$$

$$\Delta\hat{u} = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+n_u-1) \end{bmatrix}, P = \begin{bmatrix} C_{da}A_{da} \\ C_{da}{A_{da}}^2 \\ \vdots \\ C_{da}{A_{da}}^{n_y} \end{bmatrix},$$

$$P_x = \begin{bmatrix} A_{da} \\ {A_{da}}^2 \\ \vdots \\ A_{da}^{n_y} \end{bmatrix}, H_x = \begin{bmatrix} B_{da} & 0 & \cdots & 0 \\ A_{da}B_{da} & B_{da} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{da}^{n_y-1}B_{da}A_{da}^{n_y-2}B_{da}\cdots A_{da}^{n_y-n_u}B_{da} & & & \end{bmatrix},$$

$$H = \begin{bmatrix} C_{da}B_{da} & D_{da} & \cdots & 0 \\ C_{da}A_{da}B_{da} & C_{da}B_{da} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{da}A_{da}^{n_y-1}B_{da}C_{da}{A_{da}}^{n_y-2}B_{da}\cdots C_{da}{A_{da}}^{n_y-n_u}B_{da} & & & \end{bmatrix}.$$

Part II Derivation of Equation (3.7)-(3.8). The treatment for constraint conditions are as follows:

(i) Control constraint. $\underline{U} \le u(k+i) \le \overline{U}, i = 0,1,2,\cdots,n_u-1$. Take $i = 0$, the control constraints becomes $\underline{U} \le u(k) \le \overline{U}$. Subtracting $u(k-1)$ from both sides to obtain $\underline{U} - u(k-1) \le u(k) - u(k-1) \le \overline{U} - u(k-1)$. Then $\underline{U} - u(k-1) \le \Delta u(k) \le \overline{U} - u(k-1)$. Taking $k = k+1$, $\underline{U} - u(k) \le \Delta u(k+1) \le \overline{U} - u(k)$. It follows that $\underline{U} - u(k-1) \le \Delta u(k+1) + \Delta u(k) \le \overline{U} - u(k-1)$. Taking $k = k+1$, $\underline{U} - u(k-1) \le \Delta u(k+2) + \Delta u(k+1) + \Delta u(k) \le \overline{U} - u(k-1)$, and so on, $\underline{U} - u(k-1) \le \Delta u(k+n_u-1) + \cdots + \Delta u(k) \le \overline{U} - u(k-1)$. The matrix form can be expressed as

$$\begin{bmatrix} I0\cdots 0 \\ II\cdots 0 \\ \vdots\vdots\ddots \\ II\cdots I \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+n_u-1) \end{bmatrix} \le \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} (\overline{U} - u(k-1)),$$

$$-\begin{bmatrix} I0\cdots 0 \\ II\cdots 0 \\ \vdots\vdots\ddots \\ II\cdots I \end{bmatrix}\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u\left(k+n_u-1\right) \end{bmatrix} \leq -\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}(\underline{U}-u(k-1)).$$

Let

$$C_c=\begin{bmatrix} I0 & \cdots 0 \\ II & \cdots 0 \\ \vdots\vdots & \ddots \\ II & \cdots \end{bmatrix}, L=\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix},$$

$$\overline{d_u}=L(\overline{U}-u(k-1)), \underline{d_u}=-L(\underline{U}-u(k-1)).$$

Then the matrix form above becomes

$$C_c\Delta\hat{u} \leq \overline{d_u},$$
$$-C_c\Delta\hat{u} \leq \underline{d_u}.$$

(ii) Output constraint. It can be obtained from $\underline{Y} \leq y(k+i) \leq \overline{Y}, i=1,2,\cdots,n_y$ that

$$\underline{Y} \leq y(k+1) \leq \overline{Y},$$
$$\underline{Y} \leq y(k+2) \leq \overline{Y},$$
$$\vdots$$
$$\underline{Y} \leq y\left(k+n_y\right) \leq \overline{Y}.$$

Using the definition of $\hat{y}$ and the prediction Equation (3.3), one has

$$\begin{bmatrix} \underline{Y} \\ \underline{Y} \\ \vdots \\ \underline{Y} \end{bmatrix} - Px_a(k) \leq H\Delta\hat{u} \leq \begin{bmatrix} \overline{Y} \\ \overline{Y} \\ \vdots \\ \overline{Y} \end{bmatrix} - Px_a(k),$$

and then $M\Delta\hat{u} \leq d$, where

$$\overline{d_y}=\begin{bmatrix} \underline{Y} \\ \underline{Y} \\ \vdots \\ \underline{Y} \end{bmatrix} - Px_a(k),\ \underline{d_y}=-\begin{bmatrix} \overline{Y} \\ \overline{Y} \\ \vdots \\ \overline{Y} \end{bmatrix} + Px_a(k),\ M=\begin{bmatrix} C_c \\ -C_c \\ H \\ -H \end{bmatrix},\ d=\begin{bmatrix} \overline{d_u} \\ \underline{d_u} \\ \overline{d_y} \\ \underline{d_y} \end{bmatrix}.$$