Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(6): 10759–10777.
DOI:10.3934/math.2022601
Received: 11 November 2021
Revised: 07 March 2022
Accepted: 10 March 2022
Published: 31 March 2022

Research article

Application of ADMM to robust model predictive control problems for the turbofan aero-engine with external disturbances

Min Wang[1,2], **Jiao Teng**[1,3,*], **Lei Wang**[1,4] **and Junmei Wu**[1]

[1] School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
[2] Key Laboratory of Intelligent Control and Optimization for Industrial Equipment, Ministry of Education, Dalian University of Technology, Dalian 116024, China
[3] Key Laboratory for Computational Mathematics and Data Intelligence of Liaoning Province, Dalian University of Technology, Dalian 116024, China
[4] School of Science, Shihezi University, Shihezi 832003, China

* **Correspondence:** Email: jtengmath@163.com.

Abstract: In this paper, we investigate a class of optimal control problems for turbofan aero-engines considering external disturbances. The alternating direction method of multipliers (ADMM) is embedded in the framework of robust model predictive control (RMPC), which is not only able to reach a predetermined value of the engine fan speed, but is also developed to maintain the robustness of the engine control system. First, to consider the optimal control strategy for the worst-case scenario, this optimal control problem is formulated as a minimum-maximum convex optimization problem with constraints. Second, through a transformation technique, the problem can be equivalently described by a variational inequality, which is then transformed into a quadratic programming (QP) problem using a proximal point algorithm (PPA). Finally, the ADMM algorithm is used to solve a series of optimization subproblems based on the structural characteristics of the model. Computational examples illustrate the solution efficiency and robustness of the improved algorithm (RMPC-ADMM).

Keywords: turbofan aero-engine; robust model predictive control; alternating direction method of multipliers; proximal point algorithm; robustness
Mathematics Subject Classification: 37J45, 49J40, 90C25, 90C47, 93C05

1. Introduction

The aircraft's aero-engine is a critical component. Its operation is quite complicated, and it often fluctuates depending on environmental circumstances and operating states (e.g., maximum state, cruise state, acceleration and deceleration states, etc.). As a result, optimizing the control of the aero-engine is critical in order to make it run reliably and efficiently and to achieve the best performance. Stable-state control, transition-state control, and limit control are the three main control tasks [1]. There are various types of aero-engines, and this study focuses on turbofan aero-engines.

Engine control adopted closed-loop feedback control from classical control theory in the early 1950s, which improved control accuracy, dynamic performance, and engine performance significantly. Because it is simple to design and implement, this approach is still utilized for many engine controllers today. Classical feedback control theory, on the other hand, cannot ensure the system's stability and dynamic performance. For this reason, modern control theories such as linear quadratic optimal control, adaptive control, robust control, LPV control, and neural network control arose in the 1960s. Classical engine linear regulators are conservative and unable to handle complex systems with output constraint protection [2, 3]. In contrast, model predictive control (MPC) provides considerable advantages in handling engine operating restrictions explicitly [4, 5], simplifying the engine control system structure [6], and performing real-time rolling optimization. MPC has received a lot of attention in the engine control industry since then [7–10].

The MPC algorithm is a finite-horizon optimum control algorithm that uses a time-forward rolling optimization algorithm. Its rolling implementation can compensate in time for uncertainties caused by model mis-match, time variations, and disturbances. In order for the control to remain practically optimal, the algorithm constantly builds additional optimizations in the real system [11]. The classical finite horizon quadratic value function is used as the objective function in the analysis of MPC problems in Ref. [12]. In the linear time-invariant (LTI) system, the output variables and their corresponding output constraints are disregarded. The control input is not taken into consideration in the output of the LTI system in Ref. [13]. In Ref. [14], the ADMM algorithm is used to improve the real-time performance of the aero-engine nonlinear MPC by applying it to the MPC rolling optimization. Firstly, they introduce auxiliary variables and then introduce auxiliary variable constraints. The simulation results show that the improved method is more efficient. In summary, we try to apply the ADMM algorithm to the turbofan aero-engine problem according to the previous research on MPC.

The ADMM algorithm can solve block convex optimization problems quickly, reducing the difficulty of solving them by breaking down large-scale problems into smaller chunks. The ADMM algorithm is theoretically guaranteed to converge for any convex value function and constraints. In the last ten years, the ADMM algorithm, which was originally developed to solve variational inequalities, has become widely used in optimization computing [15–19]. Noor proposed methods for addressing general variational inequalities [20, 21], including projection, Wiener-Hopf equations, updating the solution, auxiliary principle, inertial proximal, penalty function, dynamical system and well-posedness. He [22] selects the appropriate matrix G in the framework of variational inequalities and solves the convex optimization problem with linear constraints using the PPA algorithm under G-modules. The method makes the iterative process of solving subproblems easy to solve. Ref. [23] directly applies the PPA algorithm's powerful convergence theory to the ADMM algorithm, demonstrating that the essence of the algorithm is the same. With the further study of the ADMM algorithm, its convergence has been proved. The PPA algorithm is a fundamental algorithm for computing optimally. It was first proposed by Martinet in 1970 [24], and then Rockafellar did further research [25, 26]. The PPA algorithm on Lagrangian multipliers is known as the augmented Lagrangian method (ALM) [27]. He [28] selects appropriate proximal parameters in the linear constrained convex minimization problem and solves it with the PPA algorithm. The subproblems obtained by this algorithm are easier to solve than the ones of the ALM algorithm. The PPA algorithm has received further attention in recent years, both in terms of theory and implementation [29–32].

Model uncertainty exists in industrial processes due to factors such as model mismatch and external disturbances between the practical system and the process model. These make the performance of MPC worse and even make the system unstable. Therefore, it is critical and realistic to investigate MPC's robustness [33]. Within the scope of MPC, which combines the advantages of robust control with predictive control, RMPC is one of the control methods for dealing with model uncertainty. The research mainly includes predictive control algorithm robustness analysis and this algorithm robust control [34]. There are primarily two methods for RMPC to achieve optimal control. The first is H_∞ control [35–39], while the second is min-max optimization [40–42]. The min-max optimization commonly uses the worst-case performance function. Based on invariant set theory, the method uses linear matrix inequalities (LMI) to design the RMPC controller. The optimization problem is then converted into an LMI problem in order to solve the predictive control. Previously, the system state was thought to be quantifiable online for RMPC [43–46]. However, the system state is not always measurable in practical applications. Using the state observer, Lee [47] and Mayne [48] built the output feedback RMPC (OFRMPC) with constraints in the LTI system. The state observer's parameters are fixed in this scenario, and the control inputs are optimized online by the OFRMPC. Bemporad [49] used the dynamic output feedback control method to design the OFRMPC with constraints in the LTI system.

Overall, we apply the ADMM algorithm to improve the RMPC problem's rolling optimization. To increase the solving efficiency and maintain system stability, we apply the modified algorithm to the turbofan aero-engine LTI system with external disturbances. The following are the primary innovations in this paper: (1) In this paper, we investigate a more complex model by considering the dual influence of output variables and state variables on the output values; (2) We improve the algorithm by utilizing the fact that the PPA algorithm can simplify the subproblem to transform the optimization problem; (3) We use the ADMM algorithm can take full advantage of the sparsity of the problem to optimize the algorithm and improve the solution efficiency; (4) Several transformations are applied to the original optimization problem, and the suggested method is demonstrated by an example in the RMPC problem for turbofan aero-engines.

The rest of the paper is organized as follows. In Section 2, we introduce the mathematical model of the RMPC problem for the turbofan aero-engine and its augmented form. In Section 3, we derive the RMPC prediction equations in matrix form. In Section 4, we transform the optimization problem of RMPC into a form that is suitable for solving with the ADMM algorithm. In Section 5, we calculate the comparison problem of RMPC for turbofan aero-engine in ground idling and verify the effectiveness of the RMPC-ADMM algorithm in terms of robustness and solving efficiency. In Section 6, we conclude this paper and look forward to future work.

2. RMPC model description

The model of a turbofan aero-engine with external disturbances can be described as follows:

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

where

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

The state variable x denotes the deviation of the fan speed $N_f(r/\text{min})$ and the core rotor speed $N_c(r/\text{min})$. The output variable y (also called the output constrained variable) denotes the deviation of the high-pressure compressor delivery temperature T_{48} ($^\circ R$) and the high-pressure compressor surge margin SmHPC(%). The control variable u denotes the deviation of the fuel flow rate $W_f(kg/\text{s})$, the variable stator vane angle VSV ($^\circ$), and the variable bleed valve opening VBV. A_d, B_d, C_d, and D_d denote the input and output matrices of the corresponding variables. The external disturbance variable $w(k)$ is bounded.

We now introduce the augmented state matrix $x_a(k) = \begin{bmatrix} x(k)^T & u(k-1)^T \end{bmatrix}^T$, which has the advantages of low computational effort, high computational accuracy, and strong computational stability. By applying $x_a(k)$ to (2.1), we can get the following augmented model:

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} &= \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} B_d \\ I \end{bmatrix} \Delta u(k) + \begin{bmatrix} I \\ 0 \end{bmatrix} w(k), \\ y(k) &= \begin{bmatrix} C_d & D_d \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + D_d \Delta u(k). \end{aligned} \tag{2.2}$$

The compact form can be written as:

$$\begin{aligned} x_a(k+1) &= A_{da} x_a(k) + B_{da} \Delta u(k) + F w(k), \\ y(k) &= C_{da} x_a(k) + D_{da} \Delta u(k), \end{aligned} \tag{2.3}$$

where

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

x_a denotes the augmented state variable. Δu denotes the new control variable. When solving practical problems, we usually depend on whether it is the tracking variable (ΔN_f) or the output constrained variable (ΔT_{48}, ΔSmHPC) to use different matrix coefficients. As a result, we express these two types of variables, respectively, to fulfill our practical needs. The equivalent model equation can be written as follows:

$$\begin{aligned} x_a(k+1) &= A_{da} x_a(k) + B_{da} \Delta u(k) + F w(k), \\ y(k) &= C_0 x_a(k), \\ p(k) &= C_{da} x_a(k) + D_{da} \Delta u(k), \end{aligned} \tag{2.4}$$

where $C_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$, $y(k)$ denotes the expression of ΔN_f, and $p(k)$ denotes the expression of ΔT_{48} and ΔSmHPC.

3. RMPC prediction equation

In order to reduce the number of independent variables in the optimization problem and improve the algorithm efficiency, we introduce the control horizon n_u, the prediction horizon n_y, and the control variable $\Delta u(k+i), i = 0, 1, 2, \cdots, n_y - 1$.

Since predict the control system needs control inputs throughout the prediction horizon, we make the following assumptions:

Assumption 1. *The control horizon n_u is not larger than the prediction horizon n_y, and the control variable $\Delta u(k+i)$ is constant in the later times of the control horizon n_u. That is,*

$$\begin{aligned} & n_u \le n_y, \\ & \Delta u(k+i) = 0, i = n_u, n_u+1, \cdots, n_y - 1. \end{aligned}$$

By iterating the first equation of Eq (2.4), the prediction state columns can be obtained as:

$$\begin{aligned} x_a(k+1 \mid k) &= A_{da}x_a(k) + B_{da}\Delta u(k) + Fw(k), \\ x_a(k+2 \mid k) &= A_{da}x_a(k+1 \mid k) + B_{da}\Delta u(k+1) + Fw(k+1) \\ &= A_{da}^2 x_a(k) + A_{da}B_{da}\Delta u(k) + B_{da}\Delta u(k+1) + A_{da}Fw(k) + Fw(k+1), \\ &\vdots \\ x_a(k+n_u \mid k) =& A_{da}x_a(k+n_u-1 \mid k) + B_{da}\Delta u(k+n_u-1) + Fw(k+n_u-1) \\ =& A_{da}^{n_u}x_a(k) + A_{da}^{n_u-1}B_{da}\Delta u(k) + A_{da}^{n_u-2}B_{da}\Delta u(k+1) + \cdots \\ &+ B_{da}\Delta u(k+n_u-1) + A_{da}^{n_u-1}Fw(k) + A_{da}^{n_u-2}Fw(k+1) + \cdots \\ &+ Fw(k+n_u-1), \\ &\vdots \\ x_a\left(k+n_y \mid k\right) =& A_{da}x_a\left(k+n_y-1 \mid k\right) + B_{da}\Delta u\left(k+n_y-1\right) + Fw\left(k+n_y-1\right) \\ =& A_{da}^{n_y}x_a(k) + A_{da}^{n_y-1}B_{da}\Delta u(k) + A_{da}^{n_y-2}B_{da}\Delta u(k+1) + \cdots \\ &+ A_{da}^{n_y-n_u}B_{da}\Delta u(k+n_u-1) + A_{da}^{n_y-1}Fw(k) + A_{da}^{n_y-2}Fw(k+1) + \cdots \\ &+ Fw\left(k+n_y-1\right). \end{aligned} \tag{3.1}$$

By iterating the second equation of Eq (2.4), the prediction tracking columns can be obtained as:

$$\begin{aligned} y(k+1 \mid k) &= C_0x_a(k+1 \mid k) \\ &= C_0A_{da}x_a(k) + C_0B_{da}\Delta u(k) + C_0Fw(k), \\ y(k+2 \mid k) &= C_0x_a(k+2 \mid k) \\ &= C_0{A_{da}}^2x_a(k) + C_0A_{da}B_{da}\Delta u(k) + C_0B_{da}\Delta u(k+1) + \\ & C_0A_{da}Fw(k) + C_0Fw(k+1), \\ &\vdots \\ y(k+n_u \mid k) =& C_0x_a(k+n_u \mid k) \\ =& C_0A_{da}^{n_u}x_a(k) + C_0A_{da}^{n_u-1}B_{da}\Delta u(k) + \cdots + C_0B_{da}\Delta u(k+n_u-1) + \\ & C_0A_{da}^{n_u-1}Fw(k) + C_0A_{da}^{n_u-2}Fw(k+1) + \cdots + C_0Fw(k+n_u-1), \\ &\vdots \\ y\left(k+n_y \mid k\right) =& C_0x_a\left(k+n_y \mid k\right) \\ =& C_0A_{da}^{n_y}x_a(k) + C_0A_{da}^{n_y-1}B_{da}\Delta u(k) + \cdots + C_0A_{da}^{n_y-n_u}B_{da}\Delta u(k+n_u-1) + \\ & C_0A_{da}^{n_y-1}Fw(k) + C_0A_{da}^{n_y-2}Fw(k+1) + \cdots + C_0Fw\left(k+n_y-1\right), \end{aligned} \tag{3.2}$$

where $(k+i \mid k), i = 1, \cdots, n_y$ in Eqs (3.1) and (3.2) denotes the prediction of $k+i$ times at the current time k.

Equations (3.1) and (3.2) can be simplified into matrix form as follows:

$$\hat{x}_a = E_x x_a(k) + H_x \Delta\hat{u} + G_x \hat{w},$$
$$\hat{y} = E_r x_a(k) + H_r \Delta\hat{u} + G_r \hat{w},$$

where

$$\begin{aligned}
&\hat{x}_a = \begin{bmatrix} x_a(k+1) \\ \vdots \\ x_a\left(k+n_y\right) \end{bmatrix}, \hat{y} = \begin{bmatrix} y(k+1) \\ \vdots \\ y\left(k+n_y\right) \end{bmatrix}, \Delta\hat{u} = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u\left(k+n_u-1\right) \end{bmatrix}, \\
&\hat{w} = \begin{bmatrix} w(k) \\ w(k+1) \\ \vdots \\ w\left(k+n_y-1\right) \end{bmatrix}, E_x = \begin{bmatrix} A_{da} \\ A_{da}{}^2 \\ \vdots \\ A_{da}{}^{n_y} \end{bmatrix}, H_x = \begin{bmatrix} B_{da} & 0 & 0 & \cdots \\ A_{da}B_{da} & B_{da} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ A_{da}{}^{n_y-1}B_{da} & A_{da}{}^{n_y-2}B_{da} & A_{da}^{n_y-3}B_{da} & \cdots \end{bmatrix}, \\
&E_r = \begin{bmatrix} C_0A_{da} \\ C_0A_{da}{}^2 \\ \vdots \\ C_0A_{da}{}^{n_y} \end{bmatrix}, H_r = \begin{bmatrix} C_0B_{da} & 0 & 0 & \cdots \\ C_0A_{da}B_{da} & C_0B_{da} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ C_0A_{da}{}^{n_y-1}B_{da} & C_0A_{da}{}^{n_y-2}B_{da} & C_0A_{da}{}^{n_y-3}B_{da} & \cdots \end{bmatrix}, \\
&G_r = \begin{bmatrix} C_0F & 0 & \cdots & 0 \\ C_0A_{da}F & C_0F & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_0A_{da}{}^{n_y-1}F & C_0A_{da}{}^{n_y-2}F & \cdots & C_0F \end{bmatrix}, G_x = \begin{bmatrix} F & 0 & \cdots & 0 \\ A_{da}F & F & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{da}{}^{n_y-1}F & A_{da}{}^{n_y-2}F & \cdots & F \end{bmatrix}.
\end{aligned} \tag{3.3}$$

4. RMPC optimization problem

During engine operation, it is necessary to guarantee that the fan speed reaches the specified reference value and that the control system remains stable at all times. Therefore, we design the value function to ensure that the tracking variable is close to the reference input variable and that the control variable's fluctuation is as small as possible. The value function is as follows:

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

where $e(k) = r(k) - \hat{y}(k)$, $r(k)$ denotes the reference input (i.e., the given fan speed deviation value). The first term is expressed as the sum of the squares of the differences between the tracking variable and the reference input variable in the prediction horizon n_y. a denotes the scalar weight (i.e., the weighting factor). At each prediction time, the larger the weighting factor a, the smaller the fluctuation of the control variable is. We use a consistent weighting factor to simplify the value function. The second term is expressed as the sum of the squares of the control variation during the control horizon n_u.

When solving the RMPC optimization problem, the system needs to satisfy the assumptions as follows:

Assumption 2. *The control variable u, the output constraint variable p, and the external disturbances w are maintained within the allowed range. That is,*

$$\underline{U} \leq u(k+i) \leq \overline{U}, i = 0, 1, 2, \cdots, n_u - 1, \tag{4.1}$$

$$\underline{P} \leq p(k+i) \leq \overline{P}, i = 1, 2, \cdots, n_y, \tag{4.2}$$

$$\underline{w} \leq w(k+i) \leq \overline{w}, i = 0, 1, 2, \cdots, n_y - 1, \tag{4.3}$$

where $\underline{U}, \overline{U}, \underline{P}(=\underline{Y}), \overline{P}(=\overline{Y}), \underline{w}$ and $\overline{w}$ denote the upper and lower bound vectors of the control variable, the output constraint variable, and the external disturbances, respectively.

When the disturbances of the system are bounded, we can take the maximum value of the disturbances to solve the minimum value of the optimization problem. As a result, the optimization problem can be expressed in terms of the worst-case performance function on a finite horizon.

Problem 1.

$$\begin{aligned} \min_{\Delta\hat{u}} \max_{\hat{w}} \quad & J(\Delta\hat{u}, \hat{w}) = \textstyle\sum_{i=1}^{n_y} e(k+i)^T e(k+i) + \sum_{i=0}^{n_u-1} a\Delta u(k+i)^T \Delta u(k+i) \\ s.t. \quad & \underline{U} \leq u(k+i) \leq \overline{U}, i = 0, 1, 2, \cdots, n_u - 1, \\ & \underline{P} \leq p(k+i) \leq \overline{P}, i = 1, 2, \cdots, n_y, \\ & \underline{w} \leq w(k+i) \leq \overline{w}, i = 0, 1, 2, \cdots, n_y - 1. \end{aligned}$$

The optimization variables of Problem 1 are $\Delta\hat{u}$ and $\hat{w}$, but they are not explicit in the value function, and the variables in the constraints are with respect to u, p and w. Therefore, it is necessary to unify the variables of both value function and constraints as $\Delta\hat{u}$ and $\hat{w}$. The transformations are shown as follows:

(1) For value function

$$\begin{aligned} J(\Delta\hat{u}, \hat{w}) = & \sum_{i=1}^{n_y} e(k+i)^T e(k+i) + \sum_{i=1}^{n_u-1} a\Delta u(k+i)^T \Delta u(k+i) \\ & = (r - \hat{y})^T (r - \hat{y}) + a\Delta\hat{u}^T \Delta\hat{u} \\ & = (r - E_r x_a(k) - H_r\Delta\hat{u} - G_r\hat{w})^T (r - E_r x_a(k) - H_r\Delta\hat{u} - G_r\hat{w}) + a\Delta\hat{u}^T \Delta\hat{u} \\ & = \Delta\hat{u}^T \left(H_r{}^T H_r + aI\right) \Delta\hat{u} + \left(2x_a{}^T E_r{}^T H_r - 2r^T H_r\right) \Delta\hat{u} + \Delta\hat{u}^T \left(2H_r{}^T G_r\right) \hat{w} - \\ & \quad \hat{w}^T \left(-G_r^T G_r\right) \hat{w} - \left(2r^T G_r - 2x_a{}^T E_r{}^T G_r\right) \hat{w} + J_0 \\ & = \frac{1}{2}\Delta\hat{u}^T W \Delta\hat{u} + c^T \Delta\hat{u} + \Delta\hat{u}^T M\hat{w} - \frac{1}{2}\hat{w}^T S\,\hat{w} - b^T\hat{w} + J_0 \\ & = \frac{1}{2}\Delta\hat{u}^T W \Delta\hat{u} + c^T \Delta\hat{u} + \Delta\hat{u}^T M\hat{w} - \left(\frac{1}{2}\hat{w}^T S\,\hat{w} + b^T\hat{w}\right) + J_0, \end{aligned}$$

where

$$W = 2H_r{}^T H_r + 2aI, \tag{4.4}$$

$$c = 2H_r^T E_r x_a - 2H_r^T r, \tag{4.5}$$

$$M = 2H_r^T G_r, \tag{4.6}$$

$$S = -2G_r^T G_r, \tag{4.7}$$

$$
\begin{aligned}
b &= 2G_r^T r - 2G_r^T E_r x_a, \\
J_0 &= r^T r - 2x_a^T E_r^T r + x_a^T E_r^T E_r x_a, \\
r &= \begin{bmatrix} r(k+1)^T & r(k+2)^T & \cdots & r\left(k+n_y\right)^T \end{bmatrix}^T.
\end{aligned} \tag{4.8}
$$

Since J_0 is a constant at the current time k, it can be ignored.

(2) For constraints

(i) Constraints for the variable u

$$\underline{U} \leq u(k+i) \leq \overline{U}, i = 0, 1, 2, \cdots, n_u - 1.$$

Let $i = 0$ and subtract $u(k-1)$ from both sides of the inequality simultaneously:

$$\underline{U} - u(k-1) \leq \Delta u(k) \leq \overline{U} - u(k-1). \tag{4.9}$$

Set $k = k + 1$, and then substitute $u(k) - u(k-1) = \Delta u(k)$:

$$\underline{U} - u(k-1) \leq \Delta u(k+1) + \Delta u(k) \leq \overline{U} - u(k-1).$$

Still let $k = k + 1$, and then substitute $u(k) - u(k-1) = \Delta u(k)$:

$$\underline{U} - u(k-1) \leq \Delta u(k+2) + \Delta u(k+1) + \Delta u(k) \leq \overline{U} - u(k-1).$$

Similarly, we can obtain:

$$\underline{U} - u(k-1) \leq \Delta u\left(k+n_u-1\right) + \cdots + \Delta u(k) \leq \overline{U} - u(k-1).$$

Then, it can be expressed in matrix form:

$$\begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u\left(k+n_u-1\right) \end{bmatrix} \leq \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} (\overline{U} - u(k-1)),$$

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

Let

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

$$\overline{d_u} = L(\overline{U} - u(k-1)), \underline{d_u} = -L(\underline{U} - u(k-1)). \tag{4.10}$$

Then the matrix form can be expressed as:

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

(ii) Constraints for the variable p

According to the derivation process (3.2), the prediction equation for the output constraint variable p can be obtained in the same way:

$$\hat{p} = E_c x_a(k) + H_c \Delta \hat{u} + G_c \hat{w}, \tag{4.11}$$

where

$$\hat{p} = \begin{bmatrix} p(k+1) \\ p(k+2) \\ \vdots \\ p\left(k+n_y\right) \end{bmatrix}, E_c = \begin{bmatrix} C_{da}A_{da} \\ C_{da}{A_{da}}^2 \\ \vdots \\ C_{da}A_{da}^{n_y} \end{bmatrix}, G_c = \begin{bmatrix} C_{da}F & 0 & \cdots & 0 \\ C_{da}A_{da}F & C_{da}F & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{da}A_{da}^{n_y-1}F & C_{da}{A_{da}}^{n_y-2}F & \cdots & C_{da}F \end{bmatrix},$$
$$H_c = \begin{bmatrix} C_{da}B_{da} & D_{da} & 0 & \cdots \\ C_{da}A_{da}B_{da} & C_{da}B_{da} & D_{da} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ C_{da}{A_{da}}^{n_y-1}B_{da} & C_{da}{A_{da}}^{n_y-2}B_{da} & C_{da}{A_{da}}^{n_y-3}B_{da} & \cdots \end{bmatrix}. \tag{4.12}$$

According to Eqs (4.2) and (4.12), we can obtain:

$$\hat{p}_{\min} \triangleq \begin{bmatrix} \underline{P} \\ \underline{P} \\ \vdots \\ \underline{P} \end{bmatrix} \leq \hat{p} \leq \begin{bmatrix} \overline{P} \\ \overline{P} \\ \vdots \\ \overline{P} \end{bmatrix} \triangleq \hat{p}_{\max}. \tag{4.13}$$

Substitute Eq (4.11) into Eq (4.13),

$$\hat{p}_{\min} - E_c x_a(k) - G_c \hat{w} \leq H_c \Delta \hat{u} \leq \hat{p}_{\max} - E_c x_a(k) - G_c \hat{w},$$

it can be simplified as,

$$-\underline{d_p} \leq H_c \Delta \hat{u} \leq \overline{d_p}, \tag{4.14}$$

where $\overline{d_p} = \hat{p}_{\max} - E_c x_a(k) - G_c \hat{w}$ and $\underline{d_p} = -\hat{p}_{\min} + E_c x_a(k) + G_c \hat{w}$.

In summary, the constraint can be denoted as:

$$l \triangleq \begin{bmatrix} -\underline{d_u} \\ -\underline{d_p} \end{bmatrix} \leq \Psi \Delta \hat{u} \leq \begin{bmatrix} \overline{d_u} \\ \overline{d_p} \end{bmatrix} \triangleq h,$$
$$\hat{w}_{\min} \leq \hat{w} \leq \hat{w}_{\max}, \tag{4.15}$$

where $\Psi = \begin{bmatrix} C_c \\ H_c \end{bmatrix}$.

(iii) Constraints for the variable w

For $\underline{w} \le w(k+i) \le \overline{w}, i = 0, 1, 2, \cdots, n_y - 1$, the direct deformation yields:

$$\hat{w}_{\min} \triangleq \begin{bmatrix} \underline{w} \\ \underline{w} \\ \vdots \\ \underline{w} \end{bmatrix} \le \hat{w} \le \begin{bmatrix} \overline{w} \\ \overline{w} \\ \vdots \\ \overline{w} \end{bmatrix} \triangleq \hat{w}_{\max}. \tag{4.16}$$

Based on the above transform, Problem 1 can be converted as:

Problem 2.

$$\begin{array}{ll} \min_{\Delta\hat{u}} \max_{\hat{w}} & J(\Delta\hat{u}, \hat{w}) = \theta_1(\Delta\hat{u}) + \Delta\hat{u}^T M \hat{w} - \theta_2(\hat{w}) \\ s.t. & l \le \Psi\Delta\hat{u} \le h, \\ & \hat{w}_{\min} \le \hat{w} \le \hat{w}_{\max}, \end{array} \tag{4.17}$$

where $\theta_1(\Delta\hat{u}) = \frac{1}{2}\Delta\hat{u}^T W \Delta\hat{u} + c^T\Delta\hat{u}$ *and* $\theta_2(\hat{w}) = \frac{1}{2}\hat{w}^T S\hat{w} + b^T\hat{w}$.

Solving optimization problems in the framework of variational inequalities can bring great convenience [50, 51], therefore, we transform Problem 2 into the form of variational inequalities as follows:

Problem 3. *Solve for* $(\Delta\hat{u}^*, \hat{w}^*) \in \Lambda$ *such that it satisfies:*

$$\begin{bmatrix} \Delta\hat{u} - \Delta\hat{u}^* \\ \hat{w} - \hat{w}^* \end{bmatrix}^T \begin{bmatrix} f(\Delta\hat{u}^*) + M\hat{w}^* \\ g(\hat{w}^*) - M^T\Delta\hat{u}^* \end{bmatrix} \ge 0, \forall(\Delta\hat{u}, \hat{w}) \in \Lambda, \tag{4.18}$$

where

$$\nabla\theta_1(\Delta\hat{u}) = W\Delta\hat{u} + c \overset{def}{=} f(\Delta\hat{u}),$$

$$\nabla\theta_2(\hat{w}) = S\hat{w} + b \overset{def}{=} g(\hat{w}),$$

$$\{(\Delta\hat{u}, \hat{w}) \mid l \le \Psi\Delta\hat{u} \le h, \hat{w}_{\min} \le \hat{w} \le \hat{w}_{\max}\} \overset{def}{=} \Lambda.$$

It can effectively exploit the simplicity of the objective function by applying the PPA algorithm to solve this problem. So, based on the study of linear constrained convex optimization problems in the papers [22, 28], we use the PPA algorithm to solve Problem 3.

For given $(\Delta\hat{u}^v, \hat{w}^v)$, the subproblems of $\Delta\hat{u}$ and $\hat{w}$ can be denoted as:

$$\begin{cases} \widetilde{\hat{w}}^v = \operatorname{Arg\,min}\left\{\theta_2(\hat{w}) - \hat{w}^T M^T \Delta\hat{u}^v + \frac{s_1}{2}\|\hat{w} - \hat{w}^v\|^2 \mid \hat{w}_{\min} \le \hat{w} \le \hat{w}_{\max}\right\}, \\ \Delta\widetilde{\hat{u}}^v = \operatorname{Arg\,min}\left\{\theta_1(\Delta\hat{u}) + \frac{r_1}{2}\left\|\Delta\hat{u} - \left[\Delta\hat{u}^v - \frac{1}{r_1}M\left(2\widetilde{\hat{w}}^v - \hat{w}^v\right)\right]\right\|^2 \mid l \le \Psi\Delta\hat{u} \le h\right\}, \end{cases} \tag{4.19}$$

where r_1 and s_1 are constants.

Simplifying Eq (4.19), we can obtain the following problem.

Problem 4.

$$\begin{array}{ll} \min_{\hat{w}} & y_1 = \hat{w}^T c_1 \hat{w} + d_1\hat{w} \\ s.t. & a_1\hat{w} \le b_1, \end{array} \tag{4.20}$$

$$\begin{array}{ll} \min_{\Delta\hat{u}} & y_2 = \Delta\hat{u}^T c_2 \Delta\hat{u} + d_2\Delta\hat{u} \\ s.t. & a_2\Delta\hat{u} \le b_2, \end{array} \tag{4.21}$$

where

$$\begin{aligned}
a_1 &= \begin{bmatrix} I \\ -I \end{bmatrix}, b_1 = \begin{bmatrix} \hat{w}_{\max} \\ -\hat{w}_{\min} \end{bmatrix}, \\
a_2 &= \begin{bmatrix} \Psi \\ -\Psi \end{bmatrix}, b_2 = \begin{bmatrix} h \\ -l \end{bmatrix}, \\
c_1 &= 0.5S + 0.5s_1 I, d_1 = (\Delta\hat{u}^v)^T M + b^T - s_1 (\hat{w}^v)^T, \\
c_2 &= 0.5W + 0.5r_1 I, d_2 = c^T - r_1 e_1^T, \\
e_1 &= \Delta\hat{u}^v - \frac{1}{r_1} M \left(2\widetilde{\hat{w}}^v - \hat{w}^v \right).
\end{aligned}$$

By solving Eq (4.19), we can obtain:

$$\begin{pmatrix} \Delta\hat{u} - \Delta\widetilde{\hat{u}}^v \\ \hat{w} - \widetilde{\hat{w}}^v \end{pmatrix}^T \left\{ \begin{pmatrix} f\left(\Delta\widetilde{\hat{u}}^v\right) + M\tilde{\hat{w}}^v \\ g\left(\widetilde{\hat{w}}^v\right) - M^T \Delta\widetilde{\hat{u}}^v \end{pmatrix} + \begin{pmatrix} r_1 \left(\Delta\tilde{\hat{u}}^v - \Delta\hat{u}^v\right) + M\left(\tilde{\hat{w}}^v - \hat{w}^v\right) \\ M^T \left(\Delta\widetilde{\hat{u}}^v - \Delta\hat{u}^v\right) + s_1 \left(\widetilde{\hat{w}}^v - \hat{w}^v\right) \end{pmatrix} \right\} \geq 0, \quad \forall (\Delta\hat{u}, \hat{w}) \in \Lambda. \tag{4.22}$$

It can be observed that the subproblems of Eq (4.19) are all quadratic programming problems with high-dimension block matrices, and there is a solution order among the subproblems, so the Problem 4 is hard to solve. Considering that the ADMM algorithm has the property of distributed optimization, we apply it to the rolling optimization of the 4 to increase the solution efficiency.

Firstly, we introduce the slack variable $z_1 (> 0)$ to Problem 4:

$$\begin{aligned}
\min_{\hat{w}} \quad & y_1 = \hat{w}^T c_1 \hat{w} + d_1 \hat{w} \\
\text{s.t.} \quad & a_1 \hat{w} + z_1 = b_1, (z_1 > 0).
\end{aligned}$$

Applying the ADMM algorithm yields the iteration equation:

$$\begin{cases}
\hat{w}^{q+1} = \arg\min \left\{ L_\beta \left(\hat{w}, z_1^q, \lambda_1^q \right) \right\}, \\
z_1^{q+1} = \arg\min \left\{ L_\beta \left(\hat{w}^{q+1}, z_1, \lambda_1^q \right) \mid z_1 > 0 \right\}, \\
\lambda_1^{q+1} = \lambda_1^q + \beta \left(a_1 \hat{w}^{q+1} + z_1^{q+1} - b_1 \right),
\end{cases} \tag{4.23}$$

where $\left(\hat{w}^q, z_1^q, \lambda_1^q\right)$ denotes the current iteration point. L_β denotes the augmented Lagrangian function and can be expressed as:

$$L_\beta = (\hat{w}, z_1, \lambda_1) = \hat{w}^T c_1 \hat{w} + d_1 \hat{w} + \lambda_1^T (a_1 \hat{w} + z_1 - b_1) + \frac{\beta}{2} \| a_1 \hat{w} + z_1 - b_1 \|^2. \tag{4.24}$$

Simplifying Eq (4.23) yields:

$$\begin{cases}
\hat{w}^{q+1} = -\frac{d_1^T + a_1^T \lambda_1^q + \beta a_1^T \left(z_1^q - b_1\right)}{2c_1 + \beta a_1^T a_1}, \\
z_1^{q+1} = \max \left\{ 0, -\frac{\lambda_1^q + \beta\left(a_1 \hat{w}^{q+1} - b_1\right)}{\beta} \right\}, \\
\lambda_1^{q+1} = \lambda_1^q + \beta \left(a_1 \hat{w}^{q+1} + z_1^{q+1} - b_1 \right).
\end{cases} \tag{4.25}$$

Iterate until the termination condition is satisfied. We can get the optimal solution $\widetilde{\hat{w}}^v$ of Eq (4.20). Next, we substitute $\widetilde{\hat{w}}^v$ into the expression of e_1. Similarly, the optimal solution $\Delta\tilde{\hat{u}}^v$ of Eq (4.21) can be obtained. In conclusion, we can obtain the proximal point $\left(\Delta\widetilde{\hat{u}}^v, \widetilde{\hat{w}}^v\right) \in \Lambda$.

After that, the next iteration point is updated to $\left(\Delta\hat{u}^{v+1}, \hat{w}^{v+1}\right)$. The iteration equation can be denoted as:

$$\begin{cases} \Delta\hat{u}^{v+1} = \Delta\hat{u}^{v} - \gamma\left(\Delta\hat{u}^{v} - \Delta\tilde{\hat{u}}^{v}\right), \\ \hat{w}^{v+1} = \hat{w}^{v} - \gamma\left(\hat{w}^{v} - \widetilde{\hat{w}}^{v}\right). \end{cases} \tag{4.26}$$

We determine whether the iterative process satisfies the accuracy requirements. The iteration continues if it is not fulfilled. If it is met, the solution $(\Delta\hat{u}^{*}, \hat{w}^{*})$ of Eq (4.18), which is also the optimal solution $(\Delta\hat{u}, \hat{w})$ of Eq (4.17), can be achieved.

According to the system model Eq (2.4), the state variable $x(k+1)$ and the output variable $y(k+1)$ at time $k+1$ can be obtained. We continue the procedures above until the whole simulation horizon Simhor has been finished, where Simhor denotes the time range (Simhor $\le n_y \le n_u$) for solving the actual problem. In Algorithm 1, the specific solution stages are presented.

Algorithm 1 RMPC-ADMM algorithm

Input: A_d, B_d, C_d, D_d,$\overline{U}$, $\underline{U}$, $\overline{Y}$, $\underline{Y}$, $\overline{w}$, $\underline{w}$, n_u, n_y, Simhor, a, r, $\varepsilon > 0$, $\beta > 0, \gamma \in [1, 2)$ and $q = 0$.

1: Compute the matrices C_c, E_x, H_x, E_r, H_r, G_x, G_r, W, M, S, $\hat{w}_{\min}$, $\hat{w}_{\max}$, E_c, G_c, H_c and Ψ as in Eqs (3.3), (4.4), (4.6), (4.7), (4.9), (4.12), (4.15) and (4.16).

2: Initialization (for k = 0, 1, $\cdots$, Simhor).
(i) Given $x_a(0)$, $x(0)$, $u(-1)$, and let $u = u(-1)$;
(ii) Compute y as in Eq (2.4);
(iii) Compute $\overline{d_u}$, d_u, c and b as in Eqs (4.5), (4.8), (4.10) and (4.14).

3: Solve for the optimal control sequence $\Delta\hat{u}(k)$.
(i) Given $(\Delta\hat{u}^{v}, \hat{w}^{v})$;
(ii) Solve $\widetilde{\hat{w}}^{v}$: Given the initial point $\left(\hat{w}^{q}, z_1^{q}, \lambda_1^{q}\right)$, obtain $\left(\hat{w}^{q+1}, z_1^{q+1}, \lambda_1^{q+1}\right)$ by solving Eq (4.25). If the termination condition is satisfied, then $\tilde{\hat{w}}^{v} = \hat{w}^{q+1}$. Otherwise, let $\hat{w}^{q} = \hat{w}^{q+1}$, $z_1^{q} = z_1^{q+1}$, $\lambda_1^{q} = \lambda_1^{q+1}$;
(iii) Solve $\Delta\widetilde{\hat{u}}^{v}$: Substitute $\widetilde{\hat{w}}^{v}$ into $\overline{d_p}$, $\underline{d_p}$ and e_1. Calculate l and h as in Eq (4.15). If $\max\left(\max\left(abs\left(\Delta\hat{u}^{v} - \Delta\widetilde{\hat{u}}^{v}\right)\right), \max\left(abs\left(\hat{w}^{v} - \widetilde{\hat{w}}^{v}\right)\right)\right) \ge \varepsilon$ is satisfied, then the new iteration point is $\left(\Delta\hat{u}^{v+1}, \hat{w}^{v+1}\right)$. Otherwise $\Delta\hat{u}(k) = \Delta\widetilde{\hat{u}}^{v}$.

4: Compute the optimal control variable $u(k)$ at the current time k by taking the first components of $\Delta\hat{u}(k)$ and $\hat{w}$ respectively.

5: Calculate $x(k+1)$, $y(k+1)$ and $x_a(k+1)$ as in Eq (2.4). Update variables $u = \begin{bmatrix} u & u(k) \end{bmatrix}$ and $y = \begin{bmatrix} y & y(k+1) \end{bmatrix}$.

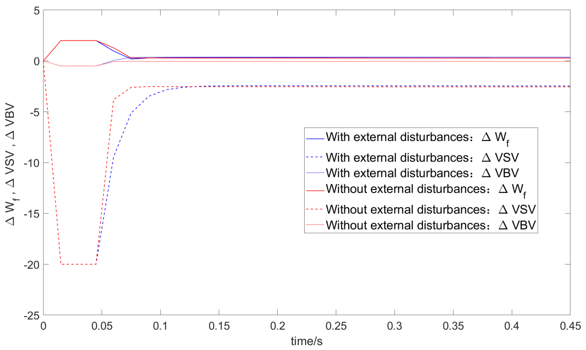
6: Let $k = k+1$ and go to step (3-iii) until the whole simulation horizon Simhor is finished.

5. Simulation example

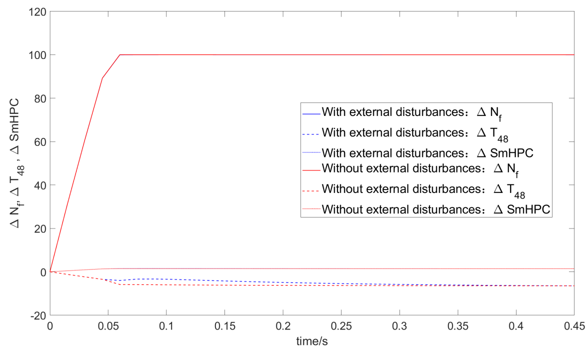
The simulation example studies the CMAPSS-40k turbofan aero-engine in ground idling. We want to reach the required fan speed response ΔN_f = 100r/min in the shortest time possible when the control variable and the output constraint variable are kept within the permitted range. For the sake of simplicity, we choose $\Delta N_f = 100\text{r/min}$, $n_u = 3$, $n_y = 7$, Simhor $= 30$. The remaining values of variables(A_d, B_d, C_d, D_d,$\overline{U}$, $\underline{U}$, $\overline{Y}$, $\underline{Y}$, $\overline{w}$, $\underline{w}$) are all consistent with the ones used in [4].

5.1. *Comparison of the results of MPC-ADMM algorithm and RMPC-ADMM algorithm*

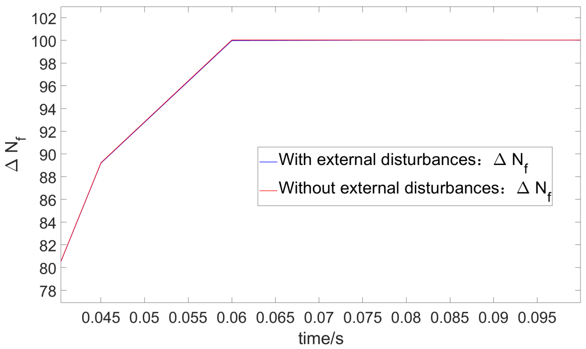
To prove the stability of the improved algorithm, we solve the two problems with and without disturbances while keeping the common parameter values constant ($\beta = 2.5$, $\gamma = 1.5$, $a = 0.01$).



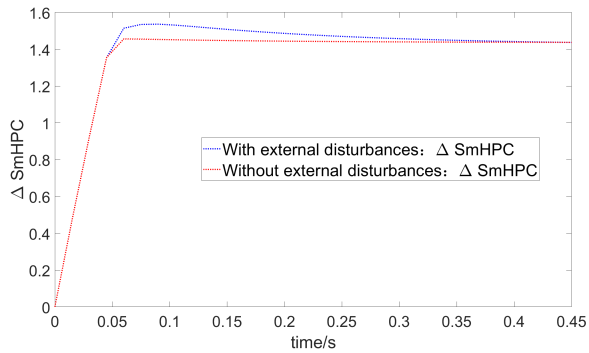
(a) Comparison of control variables with and without disturbances



(b) Comparison of output response variables with and without disturbances



(c) Local diagram of ΔN_f with and without disturbances



(d) Local diagram of ΔSmHPC with and without disturbances

Figure 1. Results of variables for two control systems with and without disturbances.

In Figure 1(a) and 1(b), the RMPC system can reach the target tracking value when the constraints are satisfied. Furthermore, there is little difference in the fluctuation of the two system variables. In Figure 1(c), both systems are almost approaching the specified fan speed response value at the same time. The difference in surge margin and numerical fluctuation between the two systems is small in Figure 1(d). These results show that the system is always in stable condition. In conclusion, by using the improved method to solve the RMPC problem, the system can reach the target value rapidly and steadily. Section 5.1 illustrates the robustness of the RMPC-ADMM algorithm.

5.2. *Comparison of the results of RMPC-QP algorithm and RMPC-ADMM algorithm*

In this section, to prove the effectiveness of the RMPC-ADMM algorithm, we calculate the same RMPC problem by using the QP algorithm (interior point method) and the ADMM algorithm, respectively. The prediction horizon n_y and the control horizon n_u may affect the predictive control performance. Too small n_y may result in failure to satisfy stability and constraints [52], while too large may affect the dynamic characteristics [53, 54]. So we will discuss it in two cases.

Case 1. $n_u = n_y$.

Firstly, we take $n_u = n_y \leq 15$. The results are shown in Table 1.

Table 1. Solution times of the two algorithms ($n_u = n_y \leq 15$).

Number	Control horizon n_u	Prediction horizon n_y	ADMM/sec	QP/sec
1	6	6	0.8924	4.3579
2	7	7	1.0174	5.2839
3	8	8	1.0187	5.7235
4	9	9	1.8554	6.0751
5	10	10	2.9395	6.7447
6	11	11	3.4386	7.2998
7	12	12	3.6307	8.164
8	13	13	3.8529	9.9916
9	14	14	4.5728	13.0046
10	15	15	4.9668	14.4309

As can be seen in Table 1, the ADMM algorithm can reduce the solution time by more than 50% when $n_u = n_y \leq 15$. This is because the ADMM algorithm keeps the KKT coefficient matrix and penalty parameters fixed during the iterations. Therefore, the ADMM algorithm needs to calculate the matrix decomposition only once in each iteration. In contrast, the interior-point method requires the inverse or decomposition of the KKT matrix in each iteration. Assuming that the solution requires M iterations, the matrix decomposition time takes t_1, and the KKT system solution back takes t_2, then the solution time of the ADMM algorithm takes $t_1 + Mt_2$, and the solution time of the interior point method takes $M(t_1 + t_2)$. In summary, as compared to the interior-point, the ADMM algorithm applied in this paper reduces the computational effort and improves the real-time performance.

Next, we further prove the effectiveness of the improved algorithm by computing the case $n_u = n_y > 15$, and the results are shown in Table 2.

Table 2. Solution time for two algorithms ($n_u = n_y > 15$).

Number	Control horizon n_u	Prediction horizon n_y	ADMM/sec	QP/sec
1	25	25	0.8924	4.3579
2	26	26	1.0174	5.2839
3	27	27	1.0187	5.7235
4	28	28	1.8554	6.0751
5	29	29	2.9395	6.7447

Case 2. $n_u \neq n_y$.

This section proves that the improved algorithm still has good results when $n_u \neq n_y$. We assume $n_u = \frac{1}{2}n_y$, and the results are shown in Table 3.

Table 3. Solution times of the two algorithms ($n_u \neq n_y$).

Number	Control horizon n_u	Prediction horizon n_y	ADMM/sec	QP/sec
1	8	16	0.8924	4.3579
2	9	18	1.0174	5.2839
3	10	20	1.0187	5.7235
4	11	22	1.8554	6.0751
5	12	24	2.9395	6.7447
6	13	26	3.4386	7.2998
7	14	28	3.6307	8.164
8	15	30	3.8529	9.9916
9	16	32	4.5728	13.0046
10	17	34	4.9668	14.4309

6. Conclusions

In this paper, we aim to solve a practical optimization problem quickly and stably. Considering that the control system in the practical problem is always uncertain, this uncertainty is divided into external uncertainty (i.e., external disturbances) and internal uncertainty (e.g., measurement error, parameter estimation error, model mismatch, etc.). As a result, we introduce the external disturbance $w(k)$ (which is also a general method) and obtain the RMPC problem in order to solve the practical problem accurately.

We study the RMPC problem for turbofan aero-engines. We take the original optimization problem as an entry point and transform it into the variational inequality. The PPA algorithm, the ADMM algorithm, the splitting contraction algorithm, and the projection contraction algorithm can all solve the variational inequality problem, the first three of which are known to have simple mechanisms and can be used with great importance in the engineering field and related disciplines. We use the PPA algorithm to transform it into problem that is applicable for the ADMM algorithm to solve. Finally, the ADMM algorithm is applied to the rolling optimization to improve its effectiveness. The application of the RMPC-ADMM algorithm to the ground idling example demonstrates that the system can still obtain the objective value while maintaining stability. Furthermore, the time it takes to solve a problem is cut in half. The research has implications for turbofan aero-engine control optimization, and further work has to be done to increase the solution efficiency. In general, it is easier to solve the variational inequality problem using the projection contraction algorithm. However, due to the high dimensionality of this practical problem, the solution is not satisfactory. In our subsequent research, we will try to solve the problem using this algorithm. Meanwhile, we will concentrate our efforts on improving the classical ADMM algorithm and increasing its solution efficiency.

Acknowledgments

The authors sincerely thank the anonymous referees for their valuable comments and detailed suggestions, which led to considerable improvement in the article. This work was supported by the National Natural Science Foundation of China (Grant Nos.12161076, 61890920, 61890921) and the

Fundamental Research Funds for the Central Universities (Grant No. DUT21LAB301).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. S. Fan, H. Li, D. Fan, *Aeroengine control (Chinese)*, Xi'an: Northwestern Polytechnic University Press, 2008.
2. J. Csank, R. May, J. Litt, T. Guo, Control design for a generic commercial aircraft engine, *Proceedings of 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2010, 6629. https://doi.org/10.2514/6.2010-6629
3. X. Du, H. Richter, Y. Guo, Multivariable sliding mode strategy with output constraints for aeroengine propulsion control, *J. Guid. Control Dynam.*, **39** (2016), 1631–1642. https://doi.org/10.2514/1.G001802
4. H. Richter, *Advanced control of turbofan engines*, New York: Springer, 2013. https://doi.org/10.1007/978-1-4614-1171-0
5. J. Mu, D. Rees, G. Liu, Advanced controller design for aircraft gas turbine engines, *Control Eng. Pract.*, **13** (2005), 1001–1015. https://doi.org/10.1016/j.conengprac.2004.11.001
6. A. Aly, I. Atia, Neural modeling and predictive control of a small turbojet engine (sr30), *Proceedings of 10th International Energy Conversion Engineering Conference*, 2012, 4242. https://doi.org/10.2514/6.2012-4242
7. X. Du, X. Sun, Z. Wang, A. Dai, A scheduling scheme of linear model predictive controllers for turbofan engines, *IEEE Access*, **5** (2017), 24533–24541. https://doi.org/10.1109/ACCESS.2017.2764076
8. J. Seok, I. Kolmanovsky, A. Girard, Coordinated model predictive control of aircraft gas turbine engine and power system, *J. Guid. Control Dynam.*, **40** (2017), 2538–2555. https://doi.org/10.2514/1.G002562
9. H. Richter, A. Singaraju, J. Litt, Multiplexed predictive control of a large commercial turbofan engine, *J. Guid. Control Dynam.*, **31** (2008), 273–281. https://doi.org/10.2514/1.30591
10. J. Decastro, Rate-based model predictive control of turbofan engine clearance, *J. Propul. Power*, **23** (2007), 804–813. https://doi.org/10.2514/1.25846
11. H. Chen, *Model predictive control (Chinese)*, Beijing: Science Press, 2013.
12. Y. Li, Q. Zou, X. Ji, C. Zhang, K. Lu, Fast model predictive control based on adaptive alternating direction method of multipliers, *J. Chem.*, **2019** (2019), 8035204. https://doi.org/10.1155/2019/8035204
13. Y. Guo, H. Gao, H. Xing, Q. Wu, Z. Lin, Decentralized coordinated voltage control for vsc-hvdc connected wind farms based on admm, *IEEE T. Sustain. Energ.*, **10** (2019), 800–810. https://doi.org/10.1109/TSTE.2018.2848467

14. R. Shan, Q. Li, F. He, H. Feng, T. Guan, Model predictive control based on ADMM for aero-engine (Chinese), *Journal of Beijing University of Aeronautics and Astronautics*, **45** (2019), 1240–1247. https://doi.org/10.13700/j.bh.1001-5965.2018.0599

15. J. Eckstein, Y. Wang, Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives, *Pac. J. Optim.*, **11** (2015), 619–644.

16. E. Ghadimi, A. Teixeira, I. Shames, M. Johansson, Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems, *IEEE T. Automat. Contr.*, **60** (2015), 644–658. https://doi.org/10.1109/TAC.2014.2354892

17. J. Bai, J. Li, F. Xu, H. Zhang, Generalized symmetric ADMM for separable convex optimization, *Comput. Optim. Appl.*, **70** (2018), 129–170. https://doi.org/10.1007/s10589-017-9971-0

18. B. He, F. Ma, X. Yuan, Optimally linearizing the alternating direction method of multipliers for convex programming, *Comput. Optim. Appl.*, **75** (2020), 361–388. https://doi.org/10.1007/s10589-019-00152-3

19. Y. Shen, Y. Zuo, A. Yu, A partially proximal S-ADMM for separable convex optimization with linear constraints, *Appl. Numer. Math.*, **160** (2021), 65–83. https://doi.org/10.1016/j.apnum.2020.09.016

20. M. Noor, Some developments in general variational inequalities, *Appl. Math. Comput.*, **152** (2004), 199–277. https://doi.org/10.1016/S0096-3003(03)00558-7

21. M. Noor, K. Noor, M. Rassias, New trends in general variational inequalities, *Acta. Appl. Math.*, **170** (2020), 981–1064. https://doi.org/10.1007/s10440-020-00366-2

22. B. He, Y. Shen, On the convergence rate of customized proximal point algorithm for convex optimization and saddle-point problem (Chinese), *Sci. Sin. Math.*, **42** (2012), 515–525. https://doi.org/10.1360/012011-1049

23. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Boston: Now Publishers, 2011. https://doi.org/10.1561/2200000016

24. B. Martinet, Brève communication. Régularisation d'inéquations variationnelles par approximations successives, *R. I. R. O.*, **4** (1970), 154–158. https://doi.org/10.1051/m2an/197004R301541

25. R. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM. J. Control. Optim.*, **14** (1976), 877–898. https://doi.org/10.1137/0314056

26. R. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Math. Oper. Res.*, **1** (1976), 97–196. https://doi.org/10.1287/moor.1.2.97

27. J. Eckstein, D. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Math. Program.*, **55** (1992), 293–318. https://doi.org/10.1007/BF01581204

28. B. He, X. Yuan, W. Zhang, A customized proximal point algorithm for convex minimization with linear constraints, *Comput. Optim. Appl.*, **56** (2013), 559–572. https://doi.org/10.1007/s10589-013-9564-5

29. C. Ha, A generalization of the proximal point algorithm, *SIAM. J. Control Optim.*, **28** (1990), 503–512. https://doi.org/10.1137/0328029

30. X. Zhao, D. Sun, K. Toh, A Newton-CG augmented Lagrangian method for semidefinite programming, *SIAM. J. Optimiz.*, **20** (2010), 1737–1765. https://doi.org/10.1137/080718206

31. G. Gu, B. He, X. Yuan, Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach, *Comput. Optim. Appl.*, **59** (2014), 135–161. https://doi.org/10.1007/s10589-013-9616-x

32. J. Bai, H. Zhang, J. Li, A parameterized proximal point algorithm for separable convex optimization, *Optim. Lett.*, **12** (2018), 1589–1608. https://doi.org/10.1007/s11590-017-1195-9

33. M. Zeilinger, D. Raimondo, A. Domahidi, M. Morari, C. Jones, On real-time robust model predictive control, *Automatica*, **50** (2014), 683–694. https://doi.org/10.1016/j.automatica.2013.11.019

34. Y. Xi, X. Geng, H. Chen, Recent advances in research on predictive control performance (Chinese), *Control Theory and Applications*, **17** (2000), 469–475.

35. J. Wang, Z. Liu, H. Chen, S. Yu, R. Pei, H_∞ output feedback control of constrained systems via moving horizon strategy, *Acta Automatica Sinica*, **33** (2007), 1176–1181. https://doi.org/10.1360/aas-007-1176

36. D. Li, Y. Xi, Design of efficient robust model predictive controller for systems with bounded disturbances (Chinese), *Control Theory and Applications*, **26** (2009), 535–539.

37. P. Orukpe, X. Zheng, I. Jaimoukha, A. Zolotas, R. Goodall, Model predictive control based on mixed H_2/H_∞ control approach for active vibration control of railway vehicles, *Vehicle Syst. Dyn.*, **46** (2008), 151–160. https://doi.org/10.1080/00423110701882371

38. H. Huang, D. Li, Y. Xi, Synthesis of robust model predictive control based on mixed H_2/H_∞ control approach (Chinese), *Kongzhi yu Juece/Control and Decision*, **25** (2010), 1269–1272.

39. H. Huang, D. Li, Z. Lin, Y. Xi, An improved robust model predictive control design in the presence of actuator saturation, *Automatica*, **47** (2011), 861–864. https://doi.org/10.1016/j.automatica.2011.01.045

40. Y. Lee, M. Cannon, B. Kouvartakis, Extended invariance and its use in model predictive control, *Automatica*, **41** (2005), 2163–2169. https://doi.org/10.1016/j.automatica.2005.07.012

41. S. Kanev, M. Verhaegen, Robustly asymptotically stable finite-horizon MPC, *Automatica*, **42** (2006), 2189–2194. https://doi.org/10.1016/j.automatica.2006.07.011

42. F. Wang, J. Jian, A nonmonotonic hybrid algorithm for min-max problem, *Optim. Eng.*, **15** (2014), 909–925. https://doi.org/10.1007/s11081-013-9229-3

43. M. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, *Automatica*, **32** (1996), 1361–1379. https://doi.org/10.1016/0005-1098(96)00063-5

44. D. Li, Y. Xi, P. Zheng, Constrained robust feedback model predictive control for uncertain systems with polytopic description, *Int. J. Control*, **82** (2009), 1267–1274. https://doi.org/10.1080/00207170802530883

45. Y. Lu. Y. Arkun, Quasi-min–max MPC algorithms for LPV systems, *Automatica*, **36** (2000), 527–540. https://doi.org/10.1016/S0005-1098(99)00176-4

46. Z. Wan, M. Kothare, Brief an efficient off-line formulation of robust model predictive control using linear matrix inequalities, *Automatica*, **39** (2003), 837–846. https://doi.org/10.1016/S0005-1098(02)00174-7

47. Y. Lee, B. Kouvaritakis, Receding horizon output feedback control for linear systems with input saturation, *IEE Proceedings*, **148** (2001), 109–115. https://doi.org/10.1049/ip-cta:20010292

48. D. Mayne, S. Raković, R. Findeisen, F. Allgöwer, Robust output feedback model predictive control of constrained linear systems, *Automatica*, **42** (2006), 1217–1222. https://doi.org/10.1016/j.automatica.2006.03.005

49. A. Bemporad, A. Garulli, Output-feedback predictive control of constrained linear systems via set-membership state estimation, *Int. J. Control*, **73** (2000), 655–665. https://doi.org/10.1080/002071700403420

50. B. He, A uniform framework of contraction methods for convex optimization and monotone variational inequality (Chinese), *Sci. Sin. Math.*, **48** (2018), 255. https://doi.org/10.1360/N012017-00034

51. B. He, M. Xu, A general framework of contraction methods for monotone variational inequalities, *Pac. J. Optim.*, **4** (2008), 195–212.

52. R. Shridhar, D. Cooper. A novel tuning strategy for multivariable model predictive control, *ISA T.*, **36** (1997), 273–280. https://doi.org/10.1016/S0019-0578(97)00036-0

53. D. Clarke, R. Scattolini, Constrained receding-horizon predictive control, *IEE Proceedings D*, **138** (1991), 347–354. https://doi.org/10.1049/ip-d.1991.0047

54. D. Mayne, J. Rawlings, C. Rao, P. Scokaert, Constrained model predictive control: stability and optimality, *Automatica*, **36** (2000), 789–814. https://doi.org/10.1016/S0005-1098(99)00214-9

AIMS Press