Peng *et al*. *Complex Eng Syst* 2023;3:18
DOI: 10.20517/ces.2023.22

Complex Engineering Systems

Research Article

Open Access

Event-triggered model predictive tracking control of aero-engine with varying prediction horizon

Yu Peng[1], Ning Xu[2], Peng Li[1], Shuoshuo Liu[1], Xudong Zhao[1]

[1]The Key Laboratory of Intelligent Control and Optimization for Industrial Equipment, Dalian University of Technology, Dalian 116024, Shengyang, China.
[2]School of Information Science and Technology, Bohai University, Jinzhou 121013, Liaoning, China.

Correspondence to: Ning Xu, School of Information Science and Technology, Bohai University, Jinzhou 121013, Liaoning, China. E-mail: hpxuning@163.com

How to cite this article: Peng Y, Xu N, Li P, Liu S, Zhao X. Event-triggered model predictive tracking control of aero-engine with varying prediction horizon. *Complex Eng Syst* 2023;3:18. http://dx.doi.org/10.20517/ces.2023.22

Received: 3 Aug 2023 **First Decision:** **Revised:** 1 Oct 2023 **Accepted:** 18 Oct 2023 **Published:** 29 Oct 2023

Academic Editor: Jinliang Liu, Hamid Reza Karimi **Copy Editor:** Fanglin Lan **Production Editor:** Fanglin Lan

Abstract

Aero-engine is a complex thermal-mechanical system with strong nonlinearity, uncertainty, and time variation. Thus, it is crucial to design an effective controller for such a complex system to obtain the desired performances of the aero-engine. In recent years, model predictive control (MPC) has shown great potential in dealing with control problems with complex constraints of multi-variable systems, which has been applied to aero-engine control, achieving good results. Furthermore, the MPC strategy using an event-driven mechanism is good at balancing system resources and ensuring system control performances. In this paper, the problem of event-triggered MPC for aero-engine systems with bounded disturbances is studied. Firstly, an event-triggered strategy with a dynamic forced-trigger mechanism is proposed. Then, an MPC algorithm based on an event-triggered mechanism is designed. Finally, an application to the JT9D aero-engine model provided by T-MATS verifies the effectiveness of the designed algorithm. It is shown that the calculation load is significantly reduced, which proves the superiority of this method.

Keywords: Event-triggered control, aero-engine control, model predictive control, transient state

1. INTRODUCTION

An aero-engine is a complex thermal-mechanical system that is the heart of an aircraft and determines its stability, safety, and reliability. An effective controller of an aero-engine is the key to guaranteeing the normal operation of such a complex thermal-mechanical system and the safety of an aircraft. However, designing a controller for an aero-engine remains a challenge due to its strong nonlinearity and time-varying characteristics. Recently, a lot of works on aero-engine control have been carried out, for instance:[1–6]. Among them, model predictive control (MPC) has gained significant attention due to its remarkable control effectiveness.

MPC has been favored in the field of industrial process control since it was first proposed in the 1970s[7]. With its great potential in dealing with control problems with complex constraints of multi-variable systems, some interesting results about MPC have been presented[8–12]. In the meantime, due to the fact that MPC is a method based on time rolling optimization, the long prediction time domain and the uncertainty of a system make the online optimization much more complicated. A system frequently performs the rolling solution of complex optimization control problems, resulting in a heavy burden on the online calculation of a controller, which has become the main difficulty hindering the practical application of predictive control methods. Currently, reducing the computational load of online calculations is a key focus in enhancing the efficiency of MPC.

It is worth mentioning that in the 1990s, work[13] first proposed a control scheme based on an event-driven mechanism. The so-called event-triggered control means that a designer considers detailed system behavior when designing a controller and uses it as the signal to trigger control actions. When corresponding event-driven conditions are satisfied, a system is driven to carry out the next operation. In this way, system resources can be saved without affecting control performances. Hence, the combination of an event-triggered strategy and MPC named event-triggered MPC (EMPC) has the potential to reduce the frequency of solving optimization problems and save system resources without affecting control performances. Based on the above advantages, great attention has been paid to studying event-triggered strategies and various interesting results for EMPC are reported; for example, work[14] has studied an EMPC strategy for continuous-time nonlinear systems, and an event-triggered condition has been designed by measuring errors between the tracking outputs and the reference; work[15] has proposed a new aperiodic formulation of MPC for nonlinear systems and some new event-triggered conditions has been provided in which the optimal cost is not used as the Lyapunov function candidate; work[16] has proposed a robust EMPC scheme for linear time-invariant discrete-time systems by designing an event condition based on a given probability distribution of disturbances acting on the system; work[17] has proposed an adaptive threshold-based event-triggered mechanism that dynamically determines the triggering time based on real-time performance of tracking; The article[18] has designed a local controller based on the principle of event triggering, utilizing asynchronous impact control to develop a predictive controller. However, one may notice that the aero-engine model was always rarely mentioned as a complex control model, and it still lacks an effective EMPC algorithm to be applied to the control of aero-engines, which is still a challenging open issue[19].

Motivated by the above discussions, in this paper, we shall study the problem of MPC for aero-engine by designing a novel event-triggering strategy with the aim of reducing computational load while stabilizing the closed-loop system. The main contributions of this work are as follows:

1. An EMPC algorithm for tracking control of aero-engines is designed, which can reduce the computation load compared to the traditional MPC method.

2. A dynamic force-trigger mechanism is proposed in the designed EMPC algorithm, which provides a more flexible control strategy on the premise of ensuring system control performances.

The remainder of the paper is organized as follows. In Section II, the EMPC algorithm is designed in which two different trigger strategies are proposed. An application of tracking control of a JT9D aero-engine is studied in Section III to show the effectiveness of the designed EMPC algorithm. Finally, the paper is concluded in Section IV.

2. EMPC ALGORITHM DESIGN

In this section, we show the design of the EMPC algorithm, in which the two different trigger strategies are presented.

2.1. Problem formulation

Consider an aero-engine model that is given in the following form:

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

where $x(t) \in \mathbb{R}^n$ is the system state, $y \in \mathbb{R}^k$ is the output, and $u \in \mathbb{R}^m$ is the control input. A, B, C, and D are constant matrices with appropriate dimensions. The output here contains the tracking output y_t, such as the speed of the fan or the core, and the limited output y_l involving values such as the turbine inlet temperature, the fan surge margin, and others. The control input may be composed of the fuel flow, the guide blade angle, and the nozzle area. In addition, consider a pre-given reference y_r, which represents the variation of the tracking output. Specifically, we convert the nonlinear aero-engine system into a linear model using either the small perturbation method or a built-in system identification toolkit. This approach is referenced in the paper[20]. The system (1) can be discretized as

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d \Delta u(k), \\ y(k) &= C_d x(k) + D_d \Delta u(k), \end{aligned} \tag{2}$$

where A_d, B_d, C_d, and D_d are discretized matrices. Then, we introduce the extended state $x_e^T(k) = [x^T(k)\, u^T(k-1)]$, and the system (2) can be transformed as

$$\begin{aligned} x_e(k+1) &= A_{de} x_e(k) + B_{de} \Delta u(k), \\ y(k) &= C_{de} x_e(k) + D_d \Delta u(k), \end{aligned} \tag{3}$$

where $A_{de} = \begin{pmatrix} A_d & B_d \\ 0 & I \end{pmatrix}$, $B_{de} = \begin{pmatrix} B_d \\ I \end{pmatrix}$, and $C_{de} = \begin{pmatrix} C_d & D_d \end{pmatrix}$.

Then, we consider a tracking problem that can be formulated as the following optimization problem:

$$\min_{\Delta U} J = \sum_{k=1}^{N_y} \big(y_t(t+kh) - y_r(t+kh)\big)^T P \big(y_t(t+kh) - y_r(t+kh)\big) + \sum_{j=0}^{N_u-1} \Delta u(t+jh)^T Q \Delta u(t+jh), \tag{4}$$

$$\begin{aligned} \text{s.t. } u_{\min} &\le u(t+jh) \le u_{\max}, \\ y_{\min} &\le y_l(t+kh) \le y_{\max}, \end{aligned}$$

where N_y and N_u stand for the prediction horizon and the control horizon, respectively; P and Q are weight matrices; h is the sampling interval; the formulation $(t+ih)$ represents the prediction of the relevant variable after i sampling times from current instant t; the subscripts max and min denote the maximum and minimum limits of the related variable, respectively. In MPC, the reference is always given by $y_r(k+j) = y(k) + (y_{ref} - y(k))(1 - e^{-jh/\tau})$, where y_{ref} is the target value, and τ is the time constant to make the tracking reference y_r be a smooth curve.

According to the discrete system (3), the prediction of the system can be derived:

$$\begin{aligned} X(k) &= A_x x(k) + B_x \Delta U(k), \\ Y(k) &= C_y X(k) + D_y \Delta U(k), \end{aligned} \tag{5}$$

where

$$X(k) = \begin{pmatrix} x_e(k+1) \\ \vdots \\ x_e(k+N_y) \end{pmatrix}, \Delta U(k) = \begin{pmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+N_u-1) \end{pmatrix},$$

$$A_x = \begin{pmatrix} A_{de} \\ \vdots \\ A_{de}^{N_y} \end{pmatrix},$$

$$B_x = \begin{pmatrix} B_{de} & 0 & 0 \\ \vdots & \vdots & \vdots \\ A_{de}^{N_u-1}B_{de} & \cdots & B_{de} \\ \vdots & \vdots & \vdots \\ A_{de}^{N_y-1}B_{de} & \cdots & \sum_{i=1}^{N_y-N_u} A_{de}^i B_{de} \end{pmatrix},$$

$$Y(k) = \begin{pmatrix} y_r(k+1) \\ \vdots \\ y_r(k+N_y) \end{pmatrix}, C_y = \begin{pmatrix} C_{de}A_{de} \\ \vdots \\ C_{de}A_{de}^{N_y} \end{pmatrix},$$

$$D_y = \begin{pmatrix} C_{de}B_{de} & D_d & 0 \\ \vdots & \vdots & \vdots \\ C_{de}A_{de}^{N_u-1} & \cdots & C_{de}B_{de} + D_d \\ \vdots & \vdots & \vdots \\ C_{de}A_{de}^{N_y-1}B_{de} & \cdots & C_{de}\sum_{i=1}^{N_y-N_u} A_{de}^i B_{de} + D_d \end{pmatrix}.$$

Applying the system (5) to the cost function (4), one obtains that

$$\min_{\Delta U} J = (C_y x_e(k) + D_y \Delta U(k) - y_r(k))^T P (C_y x_e(k) + D_y \Delta U(k) - y_r(k)) + \Delta U(k)^T Q \Delta U(k). \tag{6}$$

Let $M(k) = C_y x_e(k) - y_r(k)$, it can be derived from (6) that

$$\min_{\Delta U} J = (M(k) + D_y \Delta U(k))^T P (M(k) + D_y \Delta U(k)) + \Delta U(k)^T Q \Delta U(k),$$

which can be further transformed as the following quadratic programming:

$$\min_{\Delta U} J = \Delta U(k)^T (C_y^T P C_y + QI) \Delta U(k) + 2M(k)^T P C_y \Delta U(k) + M(k)^T M(k). \quad (7)$$

Note that the matrices C_y and D_y should be calculated by using the corresponding rows in C and D. For example, if $C_t x = y_t$, then C_y should be derived by using C_t. Finally, the optimization problem (4) can be transformed as the quadratic programming (7) with constrains $A_i \Delta U(k) \leq b(k)$, where $A_i = (D_y, -D_y)^T$, $b(k) = (I, -I)^T$.

By applying some traditional optimal methods, such quadratic programming is easy to solve. More details can be found in [21,22]. However, we point out that when utilizing the traditional MPC method, the optimization problem needs to be solved at each sampling time, which leads to more computing load. Thus, in the following, we shall focus on the design of the event-trigger mechanism to reduce the frequency of solving the optimization problem.

2.2. Event-Trigger Mechanism Design for MPC

The main idea of the event-triggered control is to design a threshold to check whether the control strategy should be applied or the sampled information should be updated. In EMPC, the threshold is used to determine whether the optimization problem should be solved to obtain a new control input sequence.

Consider the event

$$||y_t(t) - y_r(t)|| \geq \delta,$$

where δ is the threshold. If the error between the tracking outputs and the reference exceeds the threshold, then the event is triggered and the optimization problem (4) should be solved to derive a new control input sequence. When the event is not triggered, determining the control input is a key problem in the design of EMPC. A natural way is to upload the values in the derived control input sequence one by one if the event is not triggered. For example, assume that at instant t_k, the event is triggered. By solving the optimization problem, a control input sequence $u(t_k), u(t_k + h), ..., u(t_k + (N_u - 1)h)$ is obtained and the first value $u(t_k)$ is updated to the controller. At the next sampling instant $t_k + h$, if the event is not triggered, then update the second value $u(t_k + h)$ to the controller. Repeating this operation if the event is not triggered at the next following sampling instants until the last value $u(t_k + (N_u - 1)h)$ is updated to the controller. Then, at the next control horizon, the event should be triggered by force. Thus, such a strategy contains a forced-trigger mechanism, and the maximum trigger interval is equal to the control horizon N_u. The event-trigger mechanism is presented as follows:

$$\begin{aligned} t_k &= \min\left\{t_k^*, t_{k-1} + N_u h\right\}, \\ t_k^* &= \inf\left\{t \geq t_{k-1} \middle| ||y_t(t) - y_r(t)|| \geq \delta\right\}. \end{aligned} \quad (8)$$

There is no doubt that applying such an event-trigger mechanism can reduce the frequency of solving the optimization problem while maintaining the control effect of MPC, mostly since the control input is all based on the optimized control input sequence. However, during the transient state process, the control input may not change frequently; for example, the input reaches its upper bound and maintains its maximum input value to obtain a maximum acceleration of the aero-engine. In addition, after the last value of the optimized control input sequence was updated, if the event is still not triggered at the next sampling instant, then it implies that the error between the tracking output and the reference remains in a reasonable region. Hence, the optimization problem may not necessarily be solved to refresh the control input. Here, we give another way to determine the value of the control input:

Repeat to upload the optimized control input sequence values to the controller one by one if the event is not triggered until the last value of the sequence is uploaded. If the event is still not triggered at the next sampling instant $t_k + N_u h$, then take a ZOH to maintain the last input value, i.e., $u(t_k + N_u h) = u(t_k + (N_u - 1)h)$. In this way, the optimization problem will not be solved immediately until the event is triggered. However, such a strategy certainly reduces the control effectiveness. Thus, we introduce a dynamic forced-trigger mechanism to maintain satisfactory control effectiveness. Consider the following dynamic forced-trigger interval that is related to the variation of the reference.

$$T(\delta y_r) = \begin{cases} T_1, & 0 \leq ||\delta y_r|| < \varsigma_1 \\ T_2, & \varsigma_1 \leq ||\delta y_r|| < \varsigma_2 \\ \vdots, & \vdots \\ T_n, & \varsigma_{n-1} \leq ||\delta y_r|| < \varsigma_n \end{cases}$$

where T_n are positive integers, ς_n are positive constants, and $n \in \mathbb{Z}_+$. It provides different forced-trigger intervals according to the changing trend of the reference. Then, the total event-trigger mechanism is given as follows:

$$\begin{aligned} t_k &= \min\left\{t_k^*, t_{k-1} + T(\delta y_r)h\right\}, \\ t_k^* &= \inf\left\{t \geq t_{k-1} \middle| ||y_t(t) - y_r(t)|| \geq \delta\right\}. \end{aligned} \tag{9}$$

Up to now, the EMPC strategy has been designed. The algorithm is presented in Algorithm 1.

In the following, we shall use a dual spool high bypass engine JT9D provided in T-MATS[23] to verify the effectiveness of the designed EMPC algorithm.

3. APPLICATION

The objective is to control the speed of the low-pressure turbine n_l of the JT9D aero-engine to track the pre-given reference presented in Figure 1. Note that the reference trajectory contains three different transient states: two acceleration processes and one moderating process. Assume that the aero-engine and the reference are working at the same steady state at first, then the transient state and the control process are started at the node B, i.e., 0.5 seconds. The control input is the fuel-to-air ratio r_{fa}, and the limited outputs are the temperature after the high-pressure turbine T and the fan surge margin SM_f. During the transient state process, we require that $T \leq 2150R$ and $SM_f \geq 15\%$. For comparison, both the two event-trigger mechanisms, (8) and (9), and the traditional MPC are applied.

Let us first establish a state-space model (1), where $A = \begin{pmatrix} -0.03551 & 0.2655 \\ 0.06657 & -0.4978 \end{pmatrix}$, $B = \begin{pmatrix} 0.00002104 \\ -0.00003945 \end{pmatrix}$, $C =$

Algorithm 1 Pseudo code of EMPC: Set the prediction horizon N_y, the control horizon N_u, the threshold δ, the forced-trigger interval $T(\dot{y}_r)$, and the terminal condition.

```
1: k = 0;
2: if k = 0
3:   solve the optimization problem to obtain the control input sequence u(kh), ..., u((k + N_u − 1)h);
4:   upload u(kh) to the controller;
5:   j = 0;
6:   k = k + 1;
7: else
8:   if ||y_t(kh) − y_r(kh)|| ≥ δ or j = T(ẏ_r);
9:     solve the optimization problem to obtain the control input sequence u(kh), ..., u((k + N_u − 1)h);
10:    upload u(kh) to the controller;
11:    j = 0;
12:    k = k + 1;
13:  else
14:    if j + 1 ≤ N_u
15:      upload u((k + j + 1)h) to the controller;
16:      j = j + 1;
17:      k = k + 1;
18:    else
19:      upload u((k + N_u − 1)h) to the controller;
20:      j = j + 1;
21:      k = k + 1;
22:    end if
23:  end if
24: end if
25: go to step 2 until the terminal condition is satisfied.
```

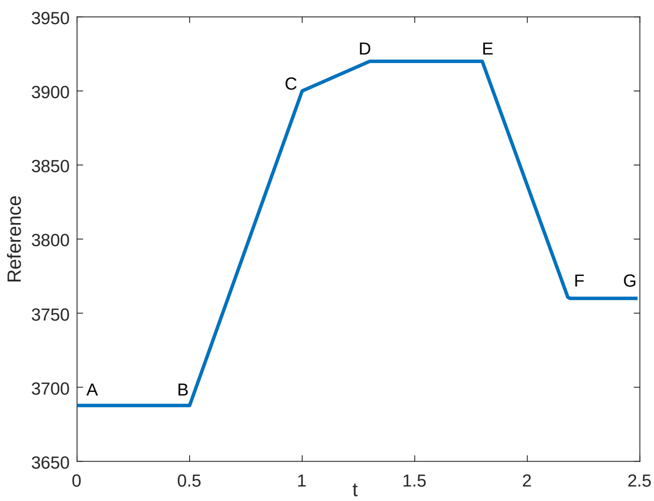

Figure 1. The reference.

$\begin{pmatrix} 0.01252 & -0.2995 \\ 0.2568 & -0.01826 \end{pmatrix}$, and $D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ [24].

The traditional MPC is first applied. The simulation step size $h = 0.01s$, and we set $N_y = N_u = 3h$. The simulation results are illustrated in Figure 2. It is obvious that the MPC method has already achieved a satisfactory control effect. The tracking error is less than 0.33%, and the limited outputs T and SM_f stay within their limits. Next, we apply the event-trigger mechanism (8) to the JT9D aero-engine under the same conditions ($h = 0.01s$, $N_y = N_u = 3h$). Setting the threshold $\delta = 1$, then the simulation results are given in Figure 3, where the tracking result and the tracking error are illustrated in (a) and (b), respectively. It is easy to see that the EMPC with (8) also yields a satisfactory control effect since the tracking result is almost the same as the MPC method. The tracking error increases only when the reference changes its working state, such as at the

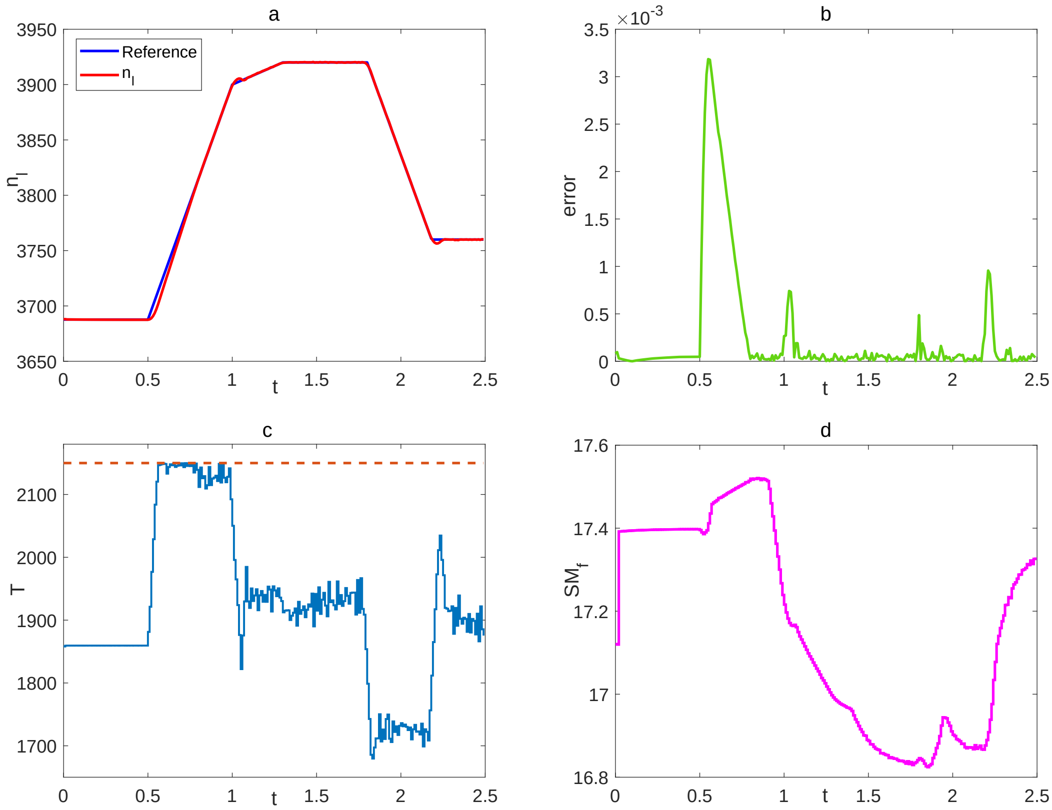

Figure 2. Simulation results for MPC method. MPC: model predictive control.

nodes B, E, and F. But it remains in a reasonable range. There is no doubt that the temperature after the high-pressure turbine T and the fan surge margin SM_f also do not exceed their restrictions during the transient state, benefiting from the strong ability of constraint management of MPC methods. Finally, the EMPC with an event-triggered mechanism (9) is utilized. Similarly, setting $h = 0.01s$, $N_y = N_u = 3h$, and $\delta = 1$. The dynamic force-trigger interval is designed as

$$T(\delta y_r) = \begin{cases} 5, & 0 \le ||\delta y_r|| < 100 \\ 10, & 100 \le ||\delta y_r|| \end{cases}$$

The main idea is to enlarge the force-trigger interval if the reference speed increases with a certain acceleration to reduce the computation load, but if the reference switches into a steady state or a small transient state, shorten the force-trigger interval to reduce the possible overshot caused by the system inertia. Then, the simulation results are shown in Figure 4. To save space, only the tracking result and the tracking error are illustrated (see Figure 4 (a) and (b), respectively). The simulations for T and SM_f are omitted here since they are almost the same as the results simulated using MPC methods. One may observe that the EMPC method with (9) also achieves a good control effect. While the tracking error is bigger than the previously used two methods when the reference changes its variation such as the nodes C, E, and F, the total tracking error is less than 0.25%. Hence, we claim that the control performance of the two designed EMPC algorithms is comparable to that of the traditional MPC method. For further comparison, we display the trigger instants generated by the three algorithms in Figure 5, where the red points stand for the force-triggered instants and the blue points represent the event-triggered instants. We point out that when implementing the MPC method, the optimization problem has been solved up to 201 times. But when applying the EMPC method, the computation load is reduced significantly: 120 times for EMPC with (8) and 80 times for EMPC with (9). Especially benefited by the dynamic force-triggered mechanism, there is no need to solve the optimization problem frequently when

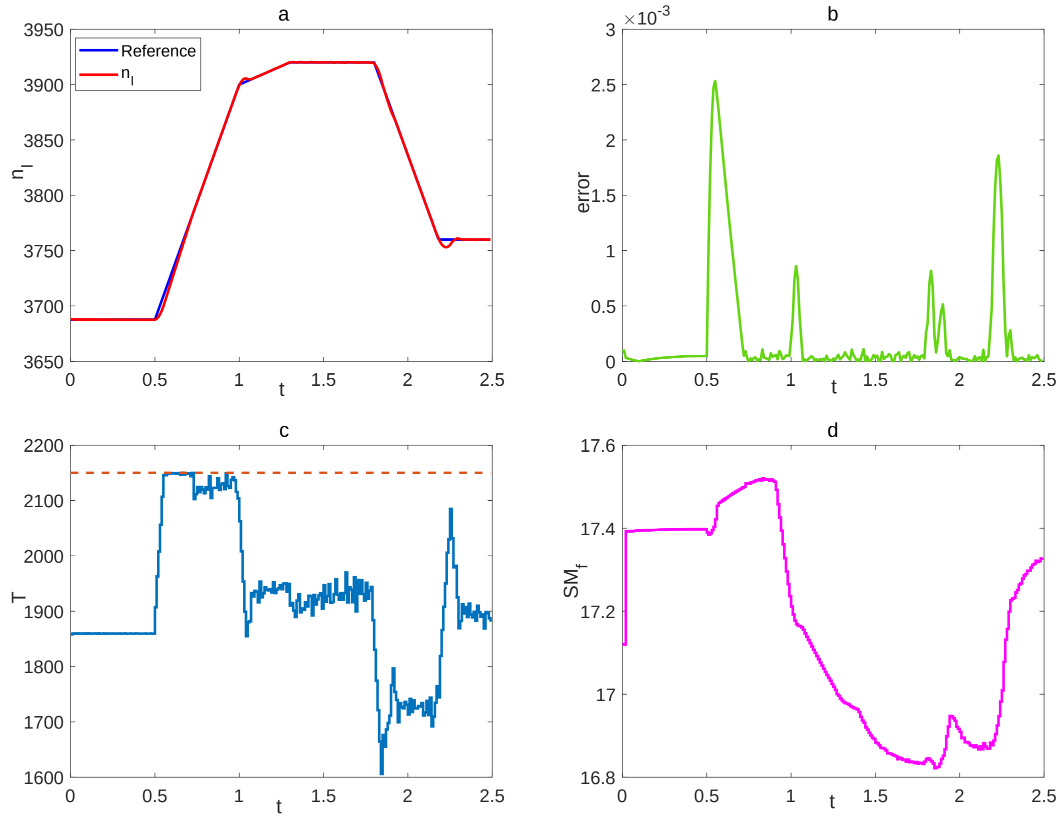

Figure 3. Simulation results for EMPC method with (8). EMPC: event-triggered model predictive control.

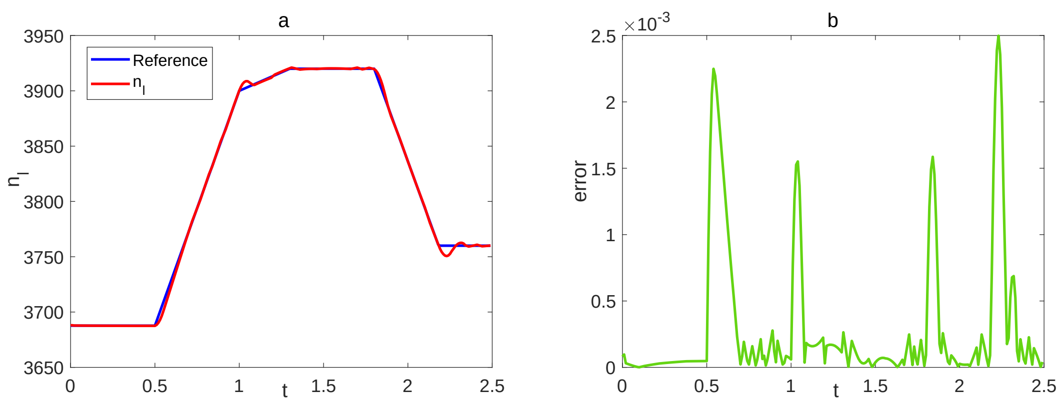

Figure 4. Simulation results for EMPC method with (9). EMPC: event-triggered model predictive control.

utilizing strategy (9). For example, during the time interval [1.2, 1.7], the optimization problem is only solved for eight times.

To summarize, the designed EMPC algorithm could reduce the computation load than the traditional MPC method. In addition, the proposed dynamic force-trigger mechanism can provide a more flexible control strategy on the premise of ensuring the control performance.

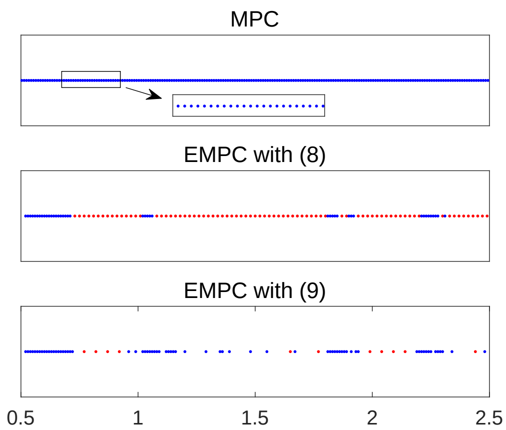

Figure 5. Trigger instants.

4. CONCLUSION

In this paper, an EMPC mechanism is proposed for aero-engine tracking, in which a dynamic forced trigger interval is designed. The tracking effect of the control method is basically guaranteed, and the frequency of solving the optimization problem is greatly reduced. This implies that the computer resources are saved. The relevant simulation experiments are carried out on the JT9D model provided by T-MATS, effectively demonstrating the theoretical results and computational efficiency of EMPC.

DECLARATIONS

Acknowledgments

The authors would like to thank the reviewers for their thoughtful comments and efforts towards improving our manuscript.

Authors' contributions

Conceptualization: Peng Y
Experiment and analyze data: Peng Y, Li P, Xu N
Manuscript drafting: Peng Y, Li Peng
Manuscript edition and review: Peng Y, Li P, Zhao X

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work is supported by the National Natural Science Foundation of China (62203064), the National Major Science and Technology Project (J2019-V-0010-0105), and the Fundamental Research Funds for the Central Universities (DUT22ZD402).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Zhou B, Xie S, Wang F, Hui J. Multi-step predictive compensated intelligent control for aero-engine wireless networked system with random scheduling. *J Franklin I* 2020;69:6154–74. DOI
2. Wang Y, Zheng Q, Xu Z, Zhang H. A novel control method for turboshaft engine with variable rotor speed based on the Ngdot estimator through LQG/LTR and rotor predicted torque feedforward. *Chinese J Aeronaut* 2020;33:1867–76. DOI
3. Cai J, Hu W, Cai K. Accuracy analysis of a single-fault Markov model for FADEC system. *J Syst Eng Electron* 2019;30:1044–52. DOI
4. Xiao L. Aeroengine multivariable nonlinear tracking control based on uncertainty and disturbance estimator. *J Eng Gas Turb Power* 2014;136:121601. DOI
5. Du X, Richter H, Guo Y. Multivariable Sliding-Mode Strategy with Output Constraints for Aeroengine Propulsion Control. *J Guid Control Dyn* 2016;39:1631-42. DOI
6. Liu T, Du X, Sun X, Richter H, Zhu F. Robust tracking control of aero-engine rotor speed based on switched LPV model. *Aerosp Sci Technol* 2019;91:382-90. DOI
7. Richalet J, Rault A, Testud JL, Papon J. Model predictive heuristic control-Applications to an industrial process. *Automatica* 1978;14:413-428. DOI
8. Gao J, Gong C, Li W, Liu J. Novel Compensation Strategy for Calculation Delay of Finite Control Set Model Predictive Current Control in PMSM. *IEEE T Ind Electron* 2020;67:5816-9. DOI
9. Zhang YQ, Yin Z, Li W, Liu J, Zhang YP. Adaptive Sliding-Mode-Based Speed Control in Finite Control Set Model Predictive Torque Control for Induction Motors. *IEEE T Ind Electron* 2021;36:8076-87. DOI
10. Zheng Q, Wang Y, Sun F, , Zhang H. Aero-engine direct thrust control with nonlinear model predictive control based on linearized deep neural network predictor. *Proc I Mech Eng* 2020;234:8076-87. DOI
11. Song S, Hei R, Ma R, Liu W. Model Predictive Control of Switched Reluctance Starter/Generator With Torque Sharing and Compensation. *IEEE T Transp Electr* 2020;6:1519-27. DOI
12. Wang Z, Du X, Sun X. The Design of Predictive Controller for Aero-engine Based on Self-tuning Model. vol. 6. ASME Turbo Expo: Turbomachinery Technical Conference and Exposition; 2019;6. DOI
13. Xi N, Tarn TJ. Intelligent planning and control for multirobot coordination:An event-based approach. *IEEE Transactions on Robotics and Automation* 1996;12:439-52. DOI
14. Li H, Shi Y. Event-triggered robust model predictive control of continuous-time nonlinear systems. *Automatica* 2014;50:1507-13. DOI
15. Hashimoto K, Adachi S, Dimarogonas DV. Event-triggered intermittent sampling for nonlinear model predictive control. *Automatica* 2017;81:148-55. DOI
16. Brunner FD, Heemels WPMH, Akkgower F. Robust Event-Triggered MPC With Guaranteed Asymptotic Bound and Average Sampling Rate. *IEEE T Automat Contr* 2017;62:5694-5709. DOI
17. Li P, Zhao X, Gou L, Qin H. Event-Triggered Model Predictive Control for Aero-Engine Transient-State Tracking Problems. *J Aerosp Inform Syst* 2023;20:5,229-39. DOI
18. Sanchez J C, Louembet C, Gavilan F, Vazquez R. An Event-Triggered Predictive Controller for Spacecraft Rendezvous Hovering Phases. *IFAC-PapersOnLine* 2019;52:97-102. DOI
19. Li P, Zhao X, Qin H, Wang Z, Niu B. Dynamic Event-Triggered Finite-Time H∞ Tracking Control of Switched LPV Aero-Engine Models. *IEEE T Circuits-II* 2022;69:1114-8. DOI
20. Hu Y, Bian Y, Gou L, Fan D. LPV/PI Control for Nonlinear Aeroengine System Based on Guardian Maps Theory. *IEEE Access* 2019;7:125854-67. DOI
21. Yu B, Li Z, Ke H, Zhang T. Wide-range model predictive control for aero-engine transient state. *Chinese J Aeronaut* 2022;35:246-60. DOI
22. Du X, Sun X, Wang Z, Dai A. A scheduling scheme of linear model predictive controllers for turbofan engines. *IEEE Access* 2017;5:24533-41. DOI
23. Jeffryes. nasa/T-MATS. Available from: https://github.com/nasa/T-MATS. [Last accessed on 21 Oct 2023].
24. Hu Y, Bian Y, Gou L, Fan D. A novel robust event-triggered MPC with state feedback for aero-engines. In: 2021 40th Chinese Control Conference (CCC), 2021. pp. 2723-30. DOI