International Journal of Innovative Computing, Information and Control
Volume **18**, Number **6**, December **2022**

ICIC International ©2022 ISSN 1349-4198
pp. 1785–1798

SYNTHESIS OF OPTIMAL CONTROLLERS FOR MODEL PREDICTIVE CONTROL

Jingwei Huang[1], Jianhong Wang[1]
and Ricardo Ambrocio Ramirez-Mendoza[2]

[1]School of Electronic Engineering and Automation
Jiangxi University of Science and Technology
No. 86, Hongqi Road, Ganzhou 341000, P. R. China
{ 9120110047; 9120180002 }@jxust.edu.cn

[2]School of Engineering and Sciences
Tecnologico de Monterrey
Monterrey, Ave. Eugenio Garza Sada 2501, Monterrey, N.L. 64849, Mexico
ricardo.ramirez@tec.mx

Received May 2022; revised August 2022

Abstract. *In this paper, synthesis analysis of optimal controllers for model predictive control is considered from two different points, i.e., theoretical analysis and practical engineering. From the aspect of theoretical analysis, optimal state feedback controllers are designed through variation analysis, corresponding to system state tracking and system output tracking. Then stability analysis for model predictive control is derived on the basis of linear matrix inequality. Model predictive control needs to solve one constraint optimization problem, being dependent of our mentioned online subgradient descent algorithm. Similarly, from the point of practical engineering, two different simulation examples are detailed shown to illustrate our proposed theoretical results.*
Keywords: Model predictive control, Optimal controllers, Stability analysis, Subgradient descent algorithm

1. **Introduction.** Model predictive control (MPC) has been developed on a model basis in the process industry area as an alternative algorithm to the conventional proportional integrate derivative (PID) control that does not utilize the model. The original version of model predictive control was developed for truncated input-output models, such as finite impulse response models or finite step response models. As it is well known, model predictive control emerges as one successful feedback strategy in many industry fields, including process industries in particular. Model predictive control is based on the conventional optimal control that is obtained by minimization or mini-maximization of some performance criterion for a fixed finite horizon or for an infinite horizon. The basic concept of model predictive control is formulated as follows. At the current time, optimal controllers for open loop or closed loop are obtained on a fixed finite horizon from the current time. Among the optimal controllers on the fixed finite horizon, only the first one is implemented as the current control law. The procedure is then repeated as the next time with a fixed finite horizon from the next time. Owing to this unique characteristic, there are several advantages for wide acceptance of model predictive control, such as closed loop structure, guaranteed stability, good tracking performance, input-output constraint handling, and simple computation. Generally, model predictive control is to construct one constraint optimization problem with input-output constraints to get the optimal controllers, solving it

DOI: 10.24507/ijicic.18.06.1785

with some numerical optimization algorithms, such as Newton algorithm, split algorithm and convex optimization algorithms.

There are vast references about the research on model predictive control from different points. More specifically, in case of the unknown but bounded noise, one bounded error identification is proposed to identify the unknown systems with time varying parameters. Then one feasible parameter set is constructed to include the unknown parameter with a given probability level. In [1], the feasible parameter set is replaced by one confidence interval, as this confidence interval can accurately describe the actual probability that the future predictor will fall into the constructed confidence interval. The problem about how to construct this confidence interval is solved by a linear approximation/programming approach [2], which can identify the unknown parameter only for linear regression model. According to the obtained feasible parameter set or confidence interval, the midpoint or center can be deemed as the final parameter estimation; further a unified framework for solving the center of the confidence interval is modified to satisfy the robustness. This robustness corresponds to other external noises, such as outlier, and unmeasured disturbance [3]. The above mentioned identification strategy, used to construct one set or interval for unknown parameter, is called as set membership identification, dealing with the unknown but bounded noise. There are two kinds of descriptions on external noise, one is probabilistic description, and the other is deterministic description, corresponding to the unknown but bounded noise here [4]. For the probabilistic description on external noise, the noise is always assumed to be one white noise, and its probabilistic density function (PDF) is known in advance. On the contrary for deterministic description on external noise, the only information about noise is bound, so this deterministic description can relax the strict assumption on probabilistic description. In reality or practice, bounded noise is more common than white noise. Within the deterministic description on external noise, set membership identification is adjusted to design controllers with two degrees of freedom [5], and it corresponds to direct data driven control or set membership control. Set membership control is applied to designing feedback control in a closed loop system with nonlinear system in [6], where the considered system is identified by set membership identification, and the obtained system parameter will be benefit for the prediction output. After substituting the obtained system parameter into the prediction output to construct one cost function, [7] takes the derivative of the above cost function with respect to control input to achieve one optimal input.

In recent years, more novel ways are explored to develop model predictive control, for example, the idea of data driven, mentioned above, is combined with model predictive control to yield a new control strategy – data driven model predictive control. In [8], data driven model predictive control is applied to designing the classical PID for a deterministic continuous time system. For the case of switching controllers in some industries, data driven model predictive control is also benefit in regulating the switching rule [9]. Consider the uncertain factors exist in the closed loop situation, one robust data driven model predictive control is proposed to alleviate and suppress the bad effect, coming from these uncertainties [10]. Further to be convenient for the use of data driven model predictive control, some existing softwares are produced for researchers, such as in python package [11] and in its intelligent form to control the heat treatment electric furnace [12]. As the number of references on model predictive control is very vast and here we cannot list all of them, so we wish the above detailed descriptions on model predictive control mean model predictive control is truly worth to be studied deeply from different points, such as theoretical analysis and practical engineering. Then the mission of this paper is to combine them together.

Based on above detailed descriptions and our previous contributions on model predictive control and its improved form, this paper shows our new derived results on model predictive control from two points, i.e., theoretical analysis and practical application. More specifically, in order to satisfy the tracking property for model predictive control, i.e., the system state and system output are all guaranteed to track their expected or desired values, on numerical optimization problem with constraint conditions is constructed to signify the tracking property. Through solving this constraint optimization problem, the optimal controller corresponds to the predictive controller. Although optimal controller and predictive control belong to different domains, i.e., optimal control and model predictive control respectively, here the obtained controller from solving that constraint optimization problem is called as the optimal controller. Consider these two different cases of tracking the desired system state and system output, variation tool and variation analysis are proposed to obtain the optimal controllers, being the state feedback controllers from our own mathematical derivations. Furthermore, as it is well known that stability is an important property for all natural system, nest stability analysis is studied for our considered model predictive control by using the tool of linear matrix inequality. If one system is unstable, then it is not any useless. When to solve that constraint optimization problem for model predictive control, one online subgradient descent algorithm is applied to generating the optimal controller, while showing our derivations on this online subgradient descent algorithm, i.e., algorithm analysis. Other than above theoretical analysis on model predictive control, the subject about applying model predictive control in numerical simulation and practical engineering is also given, so our paper studies model predictive control from theoretical analysis and practical engineering simultaneously.

This paper is organized as follows. In Section 2, preliminaries about one linear time invariant system and model predictive control are introduced and reviewed. Then Section 3 yields the optimal controllers or predictive controllers from the point of variation analysis, which correspond to the system state tracking and system output tracking. In Section 4, stability analysis is studied for model predictive control based on linear matrix inequality. Section 5 gives the detailed online subgradient descent algorithm to solve the optimal controllers. In Section 6, two simulation examples illustrate the effectiveness of the considered model predictive control from the theoretical simulation and practical engineering. Section 7 ends the paper with a final conclusion and mentions the next topic.

2. Preliminaries on Model Predictive Control.

Consider the following linear time invariant system

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \tag{1}$$

where in Equation (1), $x(k) \in R^n$, $u(k) \in R^m$, $y(k) \in R^p$ are the state, input and output of the system at the discrete time instant k, respectively. Matrices $\{A, B, C, D\}$ are four matrices with approximated dimensions.

Although linear system does not exist in our normal life, it is the nonlinear system. After linearizing the nonlinear system at some equilibrium points, one approximated linear system can be used to replace the original nonlinear form. Furthermore, our proposed theories from this paper can be also applied to the nonlinear system directly, and it is our ongoing work. Due to the fact that four matrices $\{A, B, C, D\}$ exist in above linear time invariant system (1), the problem of identifying these four matrices can be dealt with the subspace identification strategy, i.e., using the input-output data sequence $\{u(k), y(k)\}$ to yield the four matrices $\{A, B, C, D\}$ directly.

For convenience, vectors $u_{[0,t-1]}$ and $y_{[0,t-1]}$ are in vectorized forms of the time interval $[0, t-1]$, i.e.,

$$u_{[0,t-1]} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(t-1) \end{bmatrix}, \quad y_{[0,t-1]} = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(t-1) \end{bmatrix} \tag{2}$$

and $x_0 = x(t_0)$ is the original state at initial time instant t_0, and t is also one time instant.

Model predictive control is an efficient strategy for practical systems with some physical and operational constraints. Specifically, consider system (1) again, at each sampling time instant k, model predictive control solves the following numerical optimization problem to obtain the optimal control input $u(k)$.

$$\begin{aligned} &\min_{u_{[1,N]}, x_{[1,N]}} J(x,u) = \sum_{k=1}^{N} \left[(x(k) - r(k))^T Q (x(k) - r(k)) + u(k)^T R u(k) \right] \\ &\text{subjcet to } x(k+1) = Ax(k) + Bu(k) \end{aligned} \tag{3}$$

where $x(k) \in R^n$ is the current state information at time instant k, Q and R are two semi-definite weighting matrices with their suited order. $r(k)$ is a sequence of reference state strategy. This optimization problem (3) defines one quadratic tracking problem, due to its quadratic cost function. N is the total number of time instants or observed data.

Similarly, vectors $u_{[1,N]}$ and $x_{[1,N]}$ are regarded as follows.

$$u_{[1,N]} = \begin{bmatrix} u(1) \\ u(2) \\ \vdots \\ u(N) \end{bmatrix}, \quad x_{[1,N]} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} \tag{4}$$

The obvious merit of model predictive is to consider the input constraint or output constraint, then the problem of model predictive control is turned to be one numerical optimization problem, so lots of existing optimization algorithms can be used, such as Newton algorithm, gradient algorithm and our considered subgradient descent algorithm.

3. **Optimal Controller.**

3.1. **Basic form.**

For optimization problem (3), corresponding to the optimal controller for model predictive control, we give the detailed derivation for the optimization problem (3) to obtain the optimal controller $u^*(k)$. We first set a Hamiltonian function as

$$H(k) = \left[(x(k) - r(k))^T Q (x(k) - r(k)) + u(k)^T R u(k) \right] + p^T(k+1)[Ax(k) + Bu(k)] \tag{5}$$

where $k \in [1, N]$, and N is the time horizon; then we have the necessary condition for $u^*(k)$ to be an optimal controller as

$$p(k) = \frac{\partial H(k)}{\partial x(k)} = 2Q(x(k) - r(k)) + A^T p(k+1) \tag{6}$$

A necessary condition for $u(k)$ to minimize $H(k)$ is $\frac{\partial H(k)}{\partial u(k)} = 0$; thus,

$$\frac{\partial H(k)}{\partial u(k)} = 2Ru(k) + B^T p(k+1) = 0 \tag{7}$$

Since the matrix $\frac{\partial^2 H(k)}{\partial u^2(k)} = 2R$ is positive definite, and $H(k)$ is a quadratic form in $u(k)$, the optimal solution $u^*(k)$ is

$$u^*(k) = -\frac{1}{2}R^{-1}B^T p(k+1) \tag{8}$$

Assume that

$$p(k) = 2K(k)x(k) + 2g(k) \tag{9}$$

After substituting, we get

$$\begin{aligned} p(k+1) &= 2K(k+1)x(k+1) + 2g(k+1) \\ &= 2K(k+1)[Ax(k) + Bu(k)] + 2g(k+1) \\ &= 2K(k+1)\left[Ax(k) - \frac{1}{2}BR^{-1}B^T p(k+1)\right] + 2g(k+1) \end{aligned} \tag{10}$$

Solving for $p(k+1)$ yields

$$p(k+1) = \left[I + K(k+1)BR^{-1}B^T\right]^{-1}[2K(k+1)Ax(k) + 2g(k+1)] \tag{11}$$

Substituting $p(k+1)$ into Equation (8), it holds that

$$u^*(k) = -R^{-1}B^T\left[I + K(k+1)BR^{-1}B^T\right]^{-1}[K(k+1)Ax(k) + g(k+1)] \tag{12}$$

Combining Equations (6) and (10), we have

$$\begin{aligned} p(k) &= 2Q(x(k) - r(k)) + A^T\left[I + K(k+1)BR^{-1}B^T\right]^{-1}[2K(k+1)Ax(k) + 2g(k+1)] \\ &= 2\left[Q + A^T\left[I + K(k+1)BR^{-1}B^T\right]^{-1}K(k)A\right]x(k) + 2\big[-Qr(k) \\ &\quad + A^T\left[I + K(k+1)BR^{-1}B^T\right]^{-1}g(k+1)\big] \end{aligned} \tag{13}$$

Then we choose $K(k+1)$ and $g(k)$ as follows:

$$\begin{aligned} K(k) &= \left[Q + A^T\left[I + K(k+1)BR^{-1}B^T\right]^{-1}K(k)A\right] \\ &= Q + A^T\left[I + K(k+1)BR^{-1}B^T\right]^{-1}K(k+1)A \\ g(k) &= -Qr(k) + A^T\left[I + K(k+1)BR^{-1}B^T\right] \end{aligned} \tag{14}$$

For a zero reference signal, $g(k)$ becomes to be zero, and then it is simplified to

$$u^*(k) = -R^{-1}B^T\left[I + K(k+1)BR^{-1}B^T\right]^{-1}K(k+1)Ax(k) \tag{15}$$

where the optimal controller is one state feedback controller.

3.2. **Extended form.** Observing Equation (3), it is the state information $x(k)$ that exists in cost function. However, in more general or extended form, output variable $y(k)$ is to replace state information $x(k)$, i.e.,

$$J(u) = \sum_{k=1}^{N}\left[(y(k) - r(k))^T Q(y(k) - r(k)) + u(k)^T Ru(k)\right] \tag{16}$$

Substituting that output equation $y(k) = Cx(k) + Du(k)$ gets

$$\begin{aligned} &(y(k) - r(k))^T Q(y(k) - r(k)) \\ &= (Cx(k) + Du(k) - r(k))^T Q(Cx(k) + Du(k) - r(k)) \\ &= (Cx(k) - r(k))^T Q(Cx(k) - r(k)) + u^T(k)D^T QDu(k) + 2(Cx(k) - r(k))^T QDu(k) \end{aligned} \tag{17}$$

The cost function (16) is formulated as

$$J(u) = \sum_{k=1}^{N}\left[(Cx(k) + Du(k) - r(k))^T Q(Cx(k) + Du(k) - r(k))\right]$$

$$= \left[(Cx(k)-r(k))^TQ(Cx(k)-r(k))+u^T(k)D^TQDu(k)+2(Cx(k) - r(k))^TQDu(k)\right]+u(k)^TRu(k) \tag{18}$$

Similarly Hamiltonian function is set as

$$H(k) = \left[(Cx(k)-r(k))^TQ(Cx(k)-r(k))+u^T(k)D^TQDu(k)+2(Cx(k) - r(k))^TQDu(k)\right]+p^T(k+1)[Ax(k)+Bu(k)]+u(k)^TRu(k) \tag{19}$$

It holds that

$$p(k)=\frac{\partial H(k)}{\partial x(k)}=2Q(Cx(k)-r(k))C+A^Tp(k+1)+2\left(D^TQD+R\right)u(k) \tag{20}$$

A necessary condition for $u(k)$ to minimize $H(k)$ is $\frac{\partial H(k)}{\partial u(k)}=0$; thus we have

$$\frac{\partial H(k)}{\partial u(k)}=2(Cx(k)-r(k))QD+2\left(D^TQD+R\right)u(k)+B^Tp(k+1)=0 \tag{21}$$

i.e., the optimal controller is

$$u^*(k)=-\frac{1}{2}\left(D^TQD+R\right)^{-1}B^Tp(k+1)-(Cx(k)-r(k))^TQD \tag{22}$$

As state information $x(k)$ exists in the optimal controllers, we need to analyze it. Due to the state equation in system (1), we have

$$\begin{aligned} x(k) &= Ax(k-1)+Bu(k-1)=\cdots \\ &= A^kx(0)+\begin{bmatrix} A^{k-1} & A^{k-2} & \cdots & A & 1\end{bmatrix}\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(k-1)\end{bmatrix} \end{aligned} \tag{23}$$

This recursive form about state information $x(k)$ can be applied in Equation (22) to computing the optimal controller.

Comment: Observing these above two optimal controllers (15) and (22), these two optimal controllers are all state feedback controllers. Optimal controller (15) corresponds to the cost function with state information, but optimal controller (22) is for one cost function, being related with the output predictor. And the output predictor is constructed by past measured data. In later Section 4, stability analysis of the first model prediction control with the state information is exemplified.

4. **Stability Analysis.** Stability is one important issue for all system, due to the fact that unstability means useless for system. Here we only give a preliminary contribution on data driven model predictive control, and the deep detailed result about stability will be our future work.

Consider the cost function again.

$$J(x,u)=\sum_{k=1}^{N}\left[(x(k)-r(k))^TQ(x(k)-r(k))+u(k)^TRu(k)\right]$$

Our aim is to design one state feedback controller $u(k)=Hx(k)$, while minimizing the above cost function, for example, in Equation (15).

$$H=-R^{-1}B^T\left[I+K(k+1)BR^{-1}B^T\right]^{-1}K(k+1)A$$

Assume that $V(x)$ has the form

$$V(x(k))=x^T(k)Kx(k), \quad K>0 \tag{24}$$

and satisfies the following inequality

$$V(x(k+1)) - V(x(k)) \leq -\left[x^T(k)Qx(k) + u^T(k)Ru(k)\right] \tag{25}$$

As Q is one semi-definite matrix, quadratic term satisfies that $x^T(k)Qx(k) \geq 0$. After considering this to obtain

$$V(x(k+1)) - V(x(k)) \leq -\left[x^T(k)Qx(k) + u^T(k)Ru(k)\right] \leq -u^T(k)Ru(k)$$

with $u(k) = Hx(k)$, Inequality (25) is changed as

$$\begin{aligned} & x^T(k)(A+BH)^T K(A+BH)x(k) - x^T(k)Kx(k) \\ \leq & -(x(k)-r(k))^T Q(x(k)-r(k)) - x^T(k)H^T RHx(k) + x^T(k)\left(Q + H^T RH\right)x(k) \\ & + r^T(k)Qr(k) - 2x^T(k)Qr(k) \\ \leq & \ 0 \end{aligned} \tag{26}$$

The left side of Inequality (26) is expanded to be

$$\begin{aligned} & -x^T(k)Qx(k) + 2x^T(k)Qr(k) - r^T(k)Qr(k) - u^T(k)Ru(k) + x^T(k)Qx(k) \\ & + x^T(k)H^T RHx(k) - x^T(k)H^T RHx(k) + r^T(k)Qr(k) - 2x^T(k)Qr(k) \\ = & -u^T(k)Ru(k) \end{aligned}$$

Changing above Inequality (26) into one linear matrix inequality, then Equation (26) is satisfied if there exist H and K such that

$$\begin{bmatrix} (A+BH)^T K(A+BH) - K + Q + H^T RH & -Q \\ -Q & Q \end{bmatrix} \leq 0 \tag{27}$$

Applying Schur complement on Equation (27) gets

$$\begin{aligned} & (A+BH)^T K(A+BH) - K + Q + H^T RH \leq 0 \\ & (A+BH)^T K(A+BH) - K + Q + H^T RH - Q^T Q^{-1} Q \leq 0 \\ & (A+BH)^T K(A+BH) - K + Q + H^T RH - Q \leq 0 \end{aligned} \tag{28}$$

Then Equation (28) is simplified as

$$(A+BH)^T K(A+BH) - K + H^T RH \leq 0 \tag{29}$$

Continue to rewrite above Equation (29) as follows

$$-K + \begin{bmatrix} (A+BH)^T & H^T \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} (A+BH) \\ H \end{bmatrix} \leq 0 \tag{30}$$

i.e.,

$$\begin{bmatrix} -K & (A+BH) & H \\ A+BH & -K^{-1} & 0 \\ H & 0 & -R^{-1} \end{bmatrix} \leq 0 \tag{31}$$

Generally, stability about that data driven model predictive control is formulated as the following Lemma 4.1.

Lemma 4.1. *Consider the stability analysis for our considered data driven model predictive control, our mission is to find one linear optimal state feedback controller, while minimizing the constructed cost function and guaranteeing the stability, if there exist H and K such that linear matrix inequality (31) holds. Then H is the optimal controller and K is a constant matrix in the called Lyapunov function $x^T(k)Kx(k)$.*

Furthermore, that optimal state feedback controller H can be solved easily. As optimal controller H must satisfy Inequality (29), for example, we set the right part to be $-I \leq 0$.

$$\begin{aligned} &(A+BH)^T K(A+BH) - K + H^T RH = -I \\ &H^T B^T KBH + H^T RH - K + I + A^T KA + 2H^T B^T KA = 0 \end{aligned} \tag{32}$$

or

$$H^T \left[B^T KB + R\right] + 2H^T B^T KA + A^T KA - K + I = 0 \tag{33}$$

Differentiating above Equation (33) with respect to H gets

$$\left[B^T KB + R\right] H = -2B^T KA \tag{34}$$

Then that optimal state feedback controller H^* is yielded as follows:

$$H^* = -2\left[B^T KB + R\right]^{-1} B^T KA \tag{35}$$

Observing Equations (15) and (35), these two kinds of optimal state feedback controllers are similar to each other.

5. **Online Subgradient Descent Algorithm.**

5.1. **Basic algorithm.**
For that numerical optimization problem (3) with one quadratic cost function and one constraint condition, i.e., the state equation, without loss of generality, we rewrite it again as

$$\begin{aligned} &\min_{u_{[1,N]},x_{[1,N]}} J(x,u) = \sum_{k=1}^{N} \left[(x(k)-r(k))^T Q(x(k)-r(k)) + u(k)^T Ru(k)\right] \\ &\text{subjcet to } x(k+1) = Ax(k) + Bu(k);\ k = 1, 2, \ldots, N \end{aligned} \tag{36}$$

In order to simplify the exposition of applying online subgradient descent algorithm to solve above numerical optimization problem (3), we construct one corresponding Lagrangian function $L(x, u, \lambda)$ as

$$\begin{aligned} L(x,u,\lambda) = &\sum_{k=1}^{N} \left[(x(k)-r(k))^T Q(x(k)-r(k)) + u(k)^T Ru(k)\right] \\ &+ \sum_{k=1}^{N} \lambda_k [x(k+1) - Ax(k) - Bu(k)] \end{aligned} \tag{37}$$

where $\{\lambda_k\}_{k=1}^N$ are the Lagrangian multipliers, corresponding to each state equation, and (x, u, λ) are denoted as

$$u = \begin{bmatrix} u(1) \\ u(2) \\ \vdots \\ u(N) \end{bmatrix}, \quad x = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} \tag{38}$$

For convenience to show the online subgradient descent algorithm, we combine (x, u, λ) as one unknown variable v, i.e.,

$$v = [u\ x\ \lambda]^T \tag{39}$$

Then on the basis of optimization theory, our mission is to solve the optimal optimization variable v_* to guarantee that

$$v_* = \min_v L(x, u, \lambda) = \min_v L(v) \tag{40}$$

Online subgradient descent algorithm is the following recurrence.

$$v_{t+1} = \pi_v \left(v_t - \gamma_t L^{'}(v_t) \right) \tag{41}$$

where $\gamma_t > 0$ are stepsize, $\pi_v(v)$ is the standard projector on V, where V is one set, i.e., $v \in V$. $L^{'}(v)$ is a subgradient of L at v, i.e.,

$$L(w) \geq L(v) + (w - v)^T L^{'}(v) \tag{42}$$

We always assume that $\int V \neq \phi$ and that the subgradients $L^{'}(v)$ reported by the first order oracle at point $v \in V$.

5.2. **Algorithm analysis.** As subgradient operations $L^{'}(v)$ are needed in recursive form (42), here we derive some subgradients as follows:

$$\frac{\partial x(k)}{\partial u(k)} = 0; \quad \frac{\partial x(k+1)}{\partial u(k)} = B; \quad \frac{\partial L(u, x, \lambda)}{\partial u(k)} = 2Ru(k) - B\lambda_k$$
$$\frac{\partial L(u, x, \lambda)}{\partial x(k)} = 2Q(x(k) - r(k)) - A\lambda_k; \quad \frac{\partial L(u, x, \lambda)}{\partial \lambda_k} = x(k+1) - Ax(k) - Bu(k)$$

To demonstrate the merit of online subgradient descent algorithm, the following two propositions are given to achieve it.

Proposition 5.1. *After given the original value $v(0)$, the vector $e = v - \pi_v(v)$ forms an acute angle with every vector of the form $w - \pi_v(v)$, $w \in V$, i.e.,*

$$(v - \pi_v(v))^T (w - \pi_v(v)) \leq 0, \quad \forall w \in V \tag{43}$$

Proof: Let $w \in V$, and $0 \leq t \leq 1$, we have

$$\phi(t) = \|[\pi_v(v) + t(w - \pi_v(v))] - x\|_2^2 \geq \|\pi_v(v) - v\|_2^2 = \phi(0) \tag{44}$$

Then

$$0 \leq \phi^{'}(0) = 2[\pi_v(v) - v]^T (w - \pi_v(v)) \tag{45}$$

i.e.,

$$(v - \pi_v(v))^T (w - \pi_v(v)) \leq 0 \tag{46}$$

In particular

$$\begin{aligned} \|w - v\|_2^2 &= \|w - \pi_v(v)\|_2^2 + \|\pi_v(v) - v\|_2^2 + 2(\pi_v(v) - v)^T (w - \pi_v(v)) \\ &\geq \|w - \pi_v(v)\|_2^2 + \|\pi_v(v) - v\|_2^2 \end{aligned} \tag{47}$$

i.e.,

$$\|w - \pi_v(v)\|_2^2 \leq \|w - v\|_2^2 - \|\pi_v(v) - v\|_2^2, \quad \forall w \in V \tag{48}$$

It completes the proof of Proposition 5.1.

Proposition 5.2. *For subgradient descent algorithm, then for every $w \in V$, we have*

$$\gamma_t (v_t - w)^T L^{'}(v_t) \leq \frac{1}{2}\|v_t - w\|_2^2 - \frac{1}{2}\|v_{t+1} - w\|_2^2 + \frac{1}{2}\gamma_t^2 \left\| L^{'}(v_t) \right\|_2^2 \tag{49}$$

Proof: By using above Proposition 5.1, we have

$$d_{t+1} = \frac{1}{2}\|v_{t+1} - w\|_2^2; \quad d_t = \frac{1}{2}\|v_t - w\|_2^2 \tag{50}$$

Then

$$d_{t+1} \leq \frac{1}{2} \left\| [v_t - w] - \gamma_t L^{'}(v_t) \right\|_2^2 = d_t - \gamma_t (v_t - w)^T L^{'}(v_t) + \frac{1}{2} \left\| L^{'}(v_t) \right\|_2^2 \tag{51}$$

Summing up inequalities over $t = 1, 2, \ldots, N$, we get

$$\sum_{t=1}^{N} \gamma_t (L(v_t) - L(w)) \leq d_1 - d_2 + \sum_{t=1}^{N} \frac{1}{2}\gamma_t^2 \left\| L'(v_t) \right\|_2^2 \tag{52}$$

Further it holds that

$$d_1 - d_N \leq \max_{w,v \in V} \frac{1}{2} \|w - v\|_2^2 \tag{53}$$

Then we have the following upper bound in Equation (54).

$$\max_{t \in [1,N]} L(v_t) - L_* \leq \frac{\max_{w,v \in V} \frac{1}{2}\|w - v\|_2^2 + \sum_{t=1}^{N} \frac{1}{2}\gamma_t^2 \left\| L'(v_t) \right\|_2^2}{\sum_{t=1}^{N} \gamma_t} \tag{54}$$

The above inequality shows the convergence results for the online subgradient descent algorithm.

6. **Simulation Examples.** Here during this section, two simulation examples are given to correspond to the theoretical simulation and practical engineering.

6.1. **Theoretical simulation.** Firstly we consider one linear time invariant system, described as follows.

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

Assume that the system order $n = 7$ is known, from an open loop experiment, an input-output trajectory $\{u(k), y(k)\}_{k=0}^{N-1}$ of data length $N = 1000$ is measured. All system matrices are given as

$$A = \begin{bmatrix} -0.322 & 0.064 & 0.0364 & -0.9917 & 0.0003 & 0.0008 & 0 \\ 0 & 0 & 1 & 0.0037 & 0 & 0 & 0 \\ -30.64 & 0 & -3.678 & 0.0046 & -0.7333 & 0.1315 & 0 \\ 8.5396 & 0 & -0.025 & -0.476 & -0.0319 & -0.062 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 57.29 & 0 & 0 & 0 \end{bmatrix};$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 57.2958 & 0 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 & 0 & 0 & 0 \\ 57.2958 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 20.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20.2 & 0 \end{bmatrix}; \quad D = 0$$

Based on this priori known input-output trajectory $\{u(k), y(k)\}_{k=0}^{N-1}$ of data length $N = 1000$, Equation (40) is proposed to yield the output predictor during the latter time interval. Figure 1 shows the resulting output predictors as well as the true output for comparison. From Figure 1, we see the output predictors are nice, and the errors can be neglected.

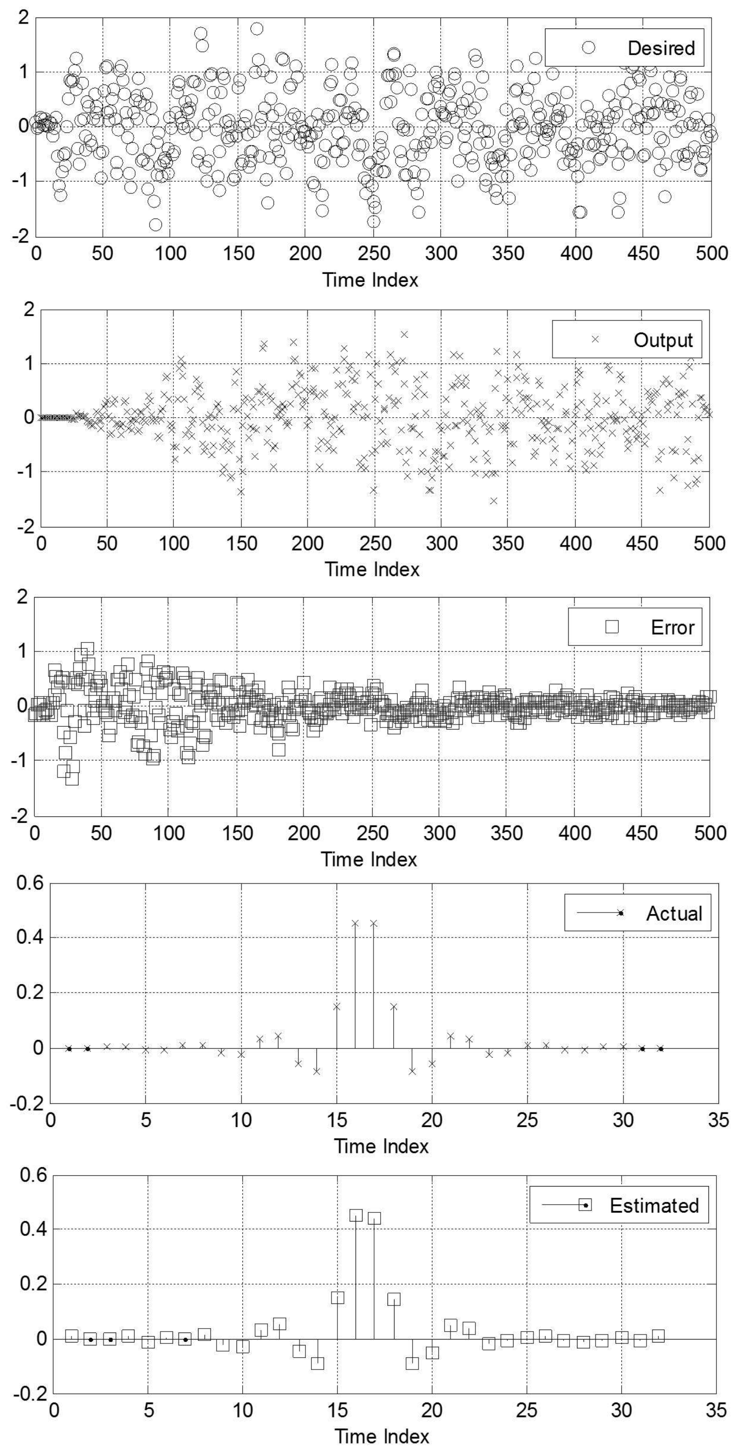


FIGURE 1. True output and estimated output

6.2. **Practical engineering.** Secondly a quadratic UAV is used, as it can take off and land vertically. It has strong air control capability, good static flight and low speed flight characteristics, which is shown in Figure 2 to achieve various flight attitudes. This quadratic UAV adopts the basic axis-symmetric layout, the four crisscrossed robots are evenly distributed, and the structure is simple. Furthermore, the quadratic UAV is mainly composed of rotor, body, flight control equipment, motor, power supply and landing gear, etc.

FIGURE 2. UAV model

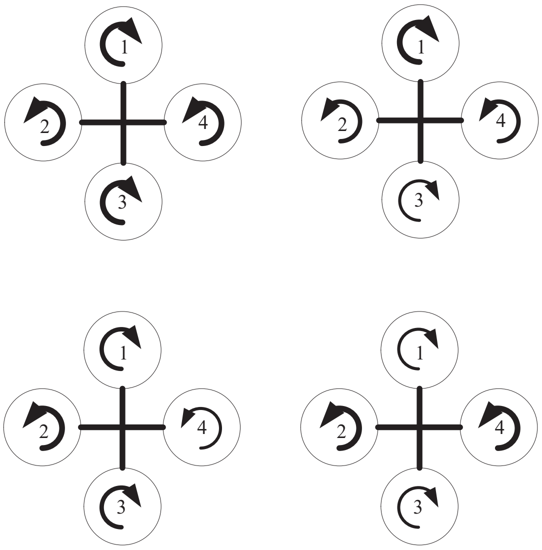


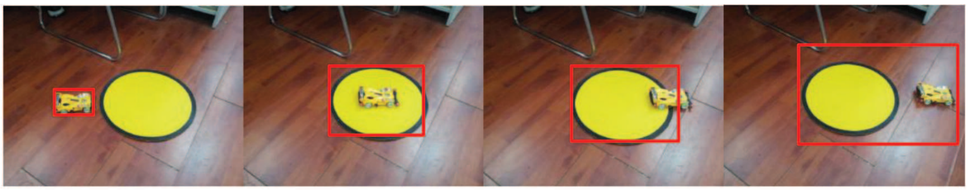
FIGURE 3. Four flight modules

The quadrotor UAV has four controllable basic motion states during flight, namely vertical flight, longitudinal flight, lateral flight and horizontal rotation. Its four flight module is plotted in Figure 3, whose vertical flight is the easiest to control among the four states. At the same time, the output power of the four motors is increased, so that the rotor speed increases and the total pulling force increases. When the total pulling force is greater than the gravity, the quadrotor flies vertically upward. The output power of the small four motors slows down the rotor speed, reduces the total pulling force, and the quadrotor flies vertically downward.

From the flight control theory in modelling the quadratic UAV, one nonlinear system is linearized at one equilibrium point, and then its approximated linear state space model is obtained. Here the linear state space model is chosen as follows:

$$A = \begin{bmatrix} -3.3978 & 0.0227 & 0.0972 \\ 0 & 0 & 1 \\ -1.0041 & 0 & -2.2879 \end{bmatrix}; \quad C = \begin{bmatrix} 3 & -2 & 1 \end{bmatrix}; \quad B = \begin{bmatrix} 0.7683 \\ 0 \\ -5.9902 \end{bmatrix}$$

To compare the tracking results of applying our proposed online subgradient descent algorithm and classical Newton algorithm, we let the quadratic UAV track one moving car, whose average velocity is set to be 0.2-5 m/s. Figure 4 shows the image sequences for three algorithms, i.e., classical Newton algorithm, improved Newton algorithm and



(a) Newton algorithm for tracking results

(b) Improved Newton algorithm for tracking results

(c) Online subgradient descent algorithm for tracking

FIGURE 4. Comparisons with Newton algorithm

online subgradient descent algorithm, and then only frames $(1, 30, 46, 53)$ are extracted from the first image sequence. In this image sequence, there is a large yellow circle with a color very similar to the four-wheel drive car as a background noise, and the car starts to move slowly and almost uniformly, and suddenly accelerates at frame 52.

After observing Figure 4 for the above Newton algorithm, when at the 17th frame, the car enters near the center of the yellow circle. Due to the interference of similar colors, the target cannot be tracked correctly. The tracking window is enlarged to include the entire yellow circle. In the 46th frame, part of the car leaves the yellow circle, and the tracking window continues to enlarge. However, for the tracking results from our online subgradient descent algorithm, through image matching, the tracking target is firstly identified and the window position and size are automatically determined. Online subgradient descent algorithm combines the advantages of the two methods to enhance the real time and robustness of tracking. No matter the target object is disturbed by similar colors or the speed is suddenly accelerated, that quadratic UAV can track the target very accurately and effectively.

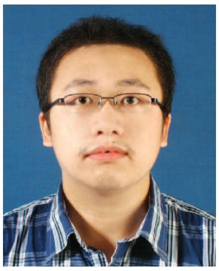
7. **Conclusion.** In this paper, synthesis analysis on model predictive control is considered from the point of optimal state feedback controller, stability and online subgradient descent algorithm. Although many subjects are mentioned, they are expanded around the idea of data driven model predictive control, whose cost function included the generated state information and output predictor from data online. As stability analysis here corresponds to our preliminary work, its deep results about stability for data driven model predictive control are our ongoing work.

Acknowledgment. This work is partially supported by Jiangxi Provinical National Science Foundation (No. GJJ180484).

REFERENCES

[1] D. Piga and S. Formentin, Direct data driven control of constrained systems, *IEEE Trans. Control Systems Technology*, vol.25, no.2, pp.331-351, 2018.
[2] M. Tanaskovic, L. Fagiano and C. Novara, Data driven control on non-line systems: An on line direct approach, *Automatica*, vol.75, no.1, pp.1-10, 2017.
[3] M. Tanaskovic, L. Fagiano and V. Gligorovski, Adaptive model predictive control for linear time varying MIMO systems, *Automatica*, vol.105, no.2, pp.237-245, 2019.
[4] E. Terzi, L. Fagiano, M. Farina and R. Scattolini, Learning based predictive control for linear systems: A unitary approach, *Automatica*, vol.108, no.1, pp.2697-2705, 2019.
[5] A. Bemporad, M. Morari and V. Dua, The explicit linear quadratic regulator for constrained systems, *Automatica*, vol.38, no.3, pp.3-20, 2002.
[6] A. Bemporad, A. Casavola and E. Mosca, Control of constrained linear systems via predictive reference management, *IEEE Trans. Automatic Control*, vol.42, no.2, pp.340-349, 1997.
[7] M. Gevers, Identification for control: From the early achievements to the revial of experiment design, *European Journal of Control*, vol.11, no.12, pp.1-10, 2017.
[8] P. Falugi and D. Q. Mayne, Getting robustness against unstructured uncertainty: A tube based MPC approach, *IEEE Trans. Automatic Control*, vol.59, no.10, pp.1290-1295, 2014.
[9] U. Rosolia and F. Borrelli, Learning model predictive control for iterative tasks: A data driven control framework, *IEEE Trans. Automatic Control*, vol.63, no.11, pp.1883-1896, 2018.
[10] E. Garone, S. Di Cairano and I. Kolmanovsky, Reference and command governors for systems with constraints: A survey on theory and applications, *Automatica*, vol.75, no.4, pp.306-328, 2017.
[11] C. Dabielson and F. Borrelli, Symmetric linear model predictive control, *IEEE Trans. Automatic Control*, vol.60, no.9, pp.1244-1259, 2015.
[12] J. Wang, Zonotope parameter identification for virtual reference feedback tuning control, *International Journal of Systems Science*, vol.50, no.7, pp.351-364, 2019.

Author Biography



Jingwei Huang received the master degree from Jiangxi University of Science and Technology, China, in 2017. He is currently a lecturer with Jiangxi University of Science and Technology, China. His research interests include machine learning and data driven control.

Jianhong Wang received the Ph.D. degree from Nanjing University of Aeronautics and Astronautics, China, in 2011. He is currently an associate professor with Jiangxi University of Science and Technology, China. His research interests include data driven control and adaptive control.

Ricardo Ambrocio Ramirez-Mendoza received a Ph.D. degree from INPG, France in 1997. He is now Professor of Mechatronics-Mechanics and Dean of Research, School of Engineering and Science at the Tecnologico de Monterrey. His main research interests include applications of advanced control to automotive systems. He is the (co-)author of 3 books, more than 80 papers in top journals and more than 200 international conference papers and more than 1,000 citations. He has worked as expert consulting for different industries and he is certified reviewer for regional development projects.