arXiv:2304.01418v1 [math.OC] 3 Apr 2023

广义数据驱动预测控制

M. Lazar 和 P. C. N. Verheijen

摘要——**数据驱动预测控制（DPC）正成为模型预测控制的一种颇具吸引力的替代方案，因为其实施所需的系统知识较少，而且智能工程系统中日益能够获取可靠数据。DPC 主要有两种方法，它们的主要区别在于预测器的构造：使用估计预测矩阵（在数据量较大时无偏），或使用汉克尔数据矩阵作为预测器（可实现偏差-方差权衡优化）。本文提出一种新颖的广义 DPC（GDPC）算法，将预测输入序列构造为已知输入序列与优化输入序列之和。与已知输入序列相对应的预测输出，采用无偏的最小二乘预测器计算；优化后的预测输出则采用基于汉克尔矩阵的预测器计算。通过结合这两类预测器，即使使用较小的汉克尔矩阵，GDPC 也能在含噪数据下实现较高性能，同时提高计算效率。文献中的一个基准示例的仿真结果表明，在存在含噪数据的情况下，使用最小尺寸汉克尔矩阵的 GDPC，其性能可与使用更大汉克尔矩阵的数据赋能预测控制相当。**

I. 引言

在现代智能工程系统中，可靠数据日益丰富，这些系统包括机电一体化、机器人学、电力电子、汽车系统和智能基础设施，参见例如 [1], [2] 及其中的参考文献。对于这些应用领域，模型预测控制（MPC） [3], [4]已成为首选的先进控制方法，原因包括约束处理、提前规划控制动作以及实现最优性能。由于获取和维护精确模型需要投入大量精力，而可靠数据在工程系统中也日益容易获得，因此开发能够在实践中实现的数据驱动预测控制（DPC）算法具有重要意义。早在 20 多年前， [5] 就提出了一种用于预测控制设计的间接数据驱动方法，即子空间预测控制（SPC）。SPC 方法无需辨识预测模型，而是利用最小二乘法从输入输出数据中辨识完整的预测矩阵。对于足够大的数据集，该方法可提供无偏预测器。

最近， [6] 基于Willems基本引理 [7],提出了一种用于预测控制设计的直接数据驱动方法，即数据赋能预测控制（DeePC）。早先， [8], 已提出使用（降阶）汉克尔矩阵作为预测器的想法，但据作者所知，第一个适定的约束数据赋能预测控制算法是在 [6],中提出的。DeePC方法完全省去了预测模型或矩阵的辨识过程，并利用由输入输出数据构造的汉克尔矩阵对预测的未来输入和输出进行参数化。在确定性、无噪声的情况下，[6], [9], 证明了模型预测控制与数据赋能预测控制等价，而[10]则证明了子空间预测控制与数据赋能预测控制等价。[9]首次通过终端等式约束以及输入输出到状态稳定的李雅普诺夫函数，为DeePC提供了稳定性保证。另一种方法是， [11] 利用终端不等式约束以及涉及储能函数与供给函数的耗散不等式，为DeePC提供稳定性保证。DeePC的一项重要进展是[12], 提出的一致正则化代价，它能在存在含噪数据时实现可靠的性能。事实上，由于DeePC算法同时求解估计与控制器综合问题， [12] 中推导的正则化使得数据受噪声污染时能够优化偏差-方差权衡。近期， [13]提出了一种系统化方法，用于调整正则化代价权重参数。

在计算方面，SPC与模型预测控制具有相同数量的优化变量，即控制输入数量乘以预测时域。在DeePC中，优化变量的数量取决于数据长度，而数据长度通常远大于预测时域。尤其是在含噪数据的情况下，为获得可靠的预测结果，需要较大的数据规模，参见例如 [9], [12], [13]。由于这会妨碍实时实现，因此提高DeePC的计算效率具有重要意义。在[14],中，通过对汉克尔数据矩阵进行LQ分解，提出了一种计算高效的DeePC形式，其在线计算复杂度与SPC/模型预测控制相同。在这种方法中，DeePC可得到与SPC类似的无偏预测器。在 [15],中，对原始汉克尔数据矩阵进行了奇异值分解，并基于所得的降阶汉克尔矩阵设计了一种DeePC算法。研究表明，这种方法能够显著降低DeePC的计算复杂度，同时提高含噪数据的预测精度。在 [16], 中，研究人员开发了一种高效的数值方法，利用汉克尔矩阵的结构求解专用于DeePC的二次规划（QP）。关于DeePC在实际应用中的使用，通常采用满足持续激励要求的最小数据规模，参见例如 [17],[18], ；或者以DeePC的无约束解代替求解二次规划，参见例如 [19]。然而，在存在含噪数据和硬约束的情况下，这些方法分别会限制可实现的性能。

本文提出一种新颖的广义数据驱动预测控制

作者来自荷兰埃因霍温理工大学电气工程系，电子邮箱：m.lazar@tue.nl, p.c.n.verheijen@tue.nl。

（GDPC）算法，该算法将已知输入序列与优化输入序列相加，构造预测输入序列。已知输入序列对应的预测输出，利用基于大型数据集的无偏最小二乘预测器计算得出。优化后的预测输出则利用基于较小（可能不同）数据集的汉克尔矩阵计算得出。根据 Willems 基本引理向多个数据集的扩展 [20], , 由两个（可能不同）数据集张成的两条轨迹之和仍是有效的系统轨迹，前提是组合后的数据矩阵整体上持续激励，且系统是线性的。通过结合这两类预测器，即使使用尺寸较小的汉克尔矩阵，GDPC 也能在含噪数据情况下实现高性能，同时保持较低的计算复杂度。本文针对 MPC 文献中的一个基准示例，评估了采用最小尺寸（根据 DeePC设计准则确定）汉克尔矩阵时 GDPC 的性能和计算复杂度，并将其与采用不同尺寸汉克尔矩阵的 DeePC 进行比较。

本文其余部分安排如下。第二节介绍必要的符号约定以及 DeePC 数据驱动预测控制方法。第三节介绍 GDPC 算法，并给出设计指南、稳定性分析及其他相关说明。第四节针对文献中的一个基准示例给出仿真结果，并与 DeePC 进行比较。第五节总结全文。

II. 预备知识

考虑一个受零均值高斯噪声 $w(k) \sim \mathcal{N}(0,\ \sigma_w^2 I)$ 影响的离散时间线性动态系统：

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \quad k \in \mathbb{N}, \\ y(k) &= Cx(k) + w(k), \end{aligned} \tag{1}$$

其中，$x \in \mathbb{R}^n$ 为状态，$u \in \mathbb{R}^{n_u}$ 为控制输入，$y \in \mathbb{R}^{n_y}$ 为测量输出；$(A,\ B,\ C)$ 为维数适当的实矩阵。假设 $(A,\ B)$ 可控，且 $(A,\ C)$ 可观。对系统 (1) 施加长度为 T 的持续激励输入序列 $\{u(k)\}_{k\in\mathbb{N}_{[0,T]}}$，即可得到相应的输出序列 $\{y(k)\}_{k\in\mathbb{N}_{[0,T]}}$。

若考虑与（1）相对应的输入输出模型，则需要引入参数 T_{ini}，以限制计算当前输出所需的过去输入输出数据窗口，即：

$$y(k) = \sum_{i=1}^{T_{\text{ini}}} a_i y(k-i) + \sum_{i=1}^{T_{\text{ini}}} b_i u(k-i), \tag{2}$$

其中系数为实数。为简化表述，我们假设输入和输出采用相同的 T_{ini}。

接下来，我们引入一些辅助记号。对于任意有限个 $q \in \mathbb{N}_{\geq 1}$ 向量 $\{\xi_1, \ldots, \xi_q\} \in \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_q}$，我们将使用算子 $\text{col}(\xi_1, \ldots, \xi_q) := [\xi_1^\top, \ldots, \xi_q^\top]^\top$。对于任意 $k \geq 0$（数据向量的起始时刻）和数据向量）以及 $j \geq 1$（数据向量的长度），定义

$$\begin{aligned} \bar{\mathbf{u}}(k, j) &:= \text{col}(u(k), \ldots, u(k+j-1)), \\ \bar{\mathbf{y}}(k, j) &:= \text{col}(y(k), \ldots, y(k+j-1)). \end{aligned}$$

令 $N \geq T_{\text{ini}}$ 表示预测时域。于是可以定义汉克尔数据矩阵：

$$\begin{aligned} \mathbf{U}_p &:= \begin{bmatrix} \bar{\mathbf{u}}(0, T_{\text{ini}}) & \ldots & \bar{\mathbf{u}}(T-1, T_{\text{ini}}) \end{bmatrix}, \\ \mathbf{Y}_p &:= \begin{bmatrix} \bar{\mathbf{y}}(1, T_{\text{ini}}) & \ldots & \bar{\mathbf{y}}(T, T_{\text{ini}}) \end{bmatrix}, \\ \mathbf{U}_f &:= \begin{bmatrix} \bar{\mathbf{u}}(T_{\text{ini}}, N) & \ldots & \bar{\mathbf{u}}(T_{\text{ini}}+T-1, N) \end{bmatrix}, \\ \mathbf{Y}_f &:= \begin{bmatrix} \bar{\mathbf{y}}(T_{\text{ini}}+1, N) & \ldots & \bar{\mathbf{y}}(T_{\text{ini}}+T, N) \end{bmatrix}. \end{aligned} \tag{3}$$

根据 DeePC 设计 [6], , 必须选择 $T_{\text{ini}} \geq n$ 和 $T \geq (n_u+1)\ (T_{\text{ini}}+N+n) - 1$，这隐含地要求对系统阶数（状态数量）作出假设。给定时刻 $k \in \mathbb{N}$ 的测量输出 $y(k)$ 以及 $T_{\text{ini}} \in \mathbb{N}_{\geq 1}$，我们定义时刻 $k \geq T_{\text{ini}}$ 已知的输入输出数据序列，它们是系统（1）的轨迹：

$$\begin{aligned} \mathbf{u}_{\text{ini}}(k) &:= \text{col}(u(k-T_{\text{ini}}), \ldots, u(k-1)), \\ \mathbf{y}_{\text{ini}}(k) &:= \text{col}(y(k-T_{\text{ini}}+1), \ldots, y(k)). \end{aligned}$$

接下来，我们定义时刻 $k \geq T_{\text{ini}}$ 的预测输入和输出序列，它们也应当是系统（1）的轨迹：

$$\begin{aligned} \mathbf{u}(k) &:= \text{col}(u(0|k), \ldots, u(N-1|k)), \\ \mathbf{y}(k) &:= \text{col}(y(1|k), \ldots, y(N|k)). \end{aligned}$$

对于正定矩阵 L，令 $L^{\frac{1}{2}}$ 表示其乔列斯基分解。在时刻 $k \geq T_{\text{ini}}$，给定 $\mathbf{u}_{\text{ini}}(k)$，$\mathbf{y}_{\text{ini}}(k)$，正则化DeePC 算法 [12] 计算如下预测输入和输出序列：

$$\begin{aligned} \min_{\mathbf{g}(k), \mathbf{u}(k), \mathbf{y}(k), \sigma(k)} \ & l_N(y(N|k)) + \sum_{i=0}^{N-1} l(y(i|k), u(i|k)) \\ & + \lambda_g l_g(\mathbf{g}(k)) + \lambda_\sigma l_\sigma(\sigma(k)) \end{aligned} \tag{4a}$$

满足以下约束：

$$\begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \\ \mathbf{Y}_f \end{bmatrix} \mathbf{g}(k) = \begin{bmatrix} \mathbf{u}_{\text{ini}}(k) \\ \mathbf{y}_{\text{ini}}(k) + \sigma(k) \\ \mathbf{u}(k) \\ \mathbf{y}(k) \end{bmatrix}, \tag{4b}$$

$$(\mathbf{y}(k), \mathbf{u}(k)) \in \mathbb{Y}^N \times \mathbb{U}^N. \tag{4c}$$

以上

$$l(y, u) := \|Q^{\frac{1}{2}}(y - r_y)\|_2^2 + \|R^{\frac{1}{2}}(u - r_u)\|_2^2, \quad l_\sigma(\sigma) := \|\sigma\|_2^2 \tag{5}$$

其中，Q，R 为一些正定矩阵。终端代价通常选得大于输出阶段代价，以确保收敛到参考值；一种常见选择是输出阶段代价的缩放版本，即 $l_N(y) := \alpha l(y,\ 0)$，$\alpha \geq 1$。参考值 $r_y \in \mathbb{R}^{n_y}$ 和 $r_u \in \mathbb{R}^{n_u}$ 可以是常数，也可以随时间变化。我们假设集合 $\mathbb{Y}$ 和 $\mathbb{U}$ 的内部中分别包含 r_y 和 r_u。该代价

$$l_g(\mathbf{g}) := \|(I - \Pi)\mathbf{g}\|_2^2, \quad \Pi := \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \end{bmatrix}, \tag{6}$$

是文献 [12], 中提出的一种正则化代价，其中 $[\cdot]^\dagger$ 表示广义伪逆。注意，使用这种正则化代价需要满足 $T \geq T_{\text{ini}}(n_u+n_y)+Nn_u$，以确保矩阵 $I-\Pi$ 具有足够大的零空间。若需要较短的数据长度 T，则也可以使用正则化代价 $l_g(\mathbf{g}) := \|\mathbf{g}\|_2^2$。然而，如 [12]所示，这种正则化并不一致。

在确定性、无噪声的情况下，DeePC 算法[6] 不需要代价项 l_g，l_σ 和变量 σ。我们观察到，DeePC 的计算复杂度主要由变量向量 $\mathbf{g} \in \mathbb{R}^T$决定，其中$T \geq (n_u+1)(T_{\text{ini}}+N+n)-1$。因此，理想情况下应采用最小的数据长度 T；但存在噪声时，为了获得准确预测 [9], [12], [13]，通常需要较大的数据长度T 。

III. 广义数据驱动预测控制

本节通过将预测输入序列 $\mathbf{u}(k)$ 构造为两个输入序列之和，提出一种新颖的广义 DPC 算法，即:

$$\mathbf{u}(k) := \bar{\mathbf{u}}(k) + \mathbf{u}_g(k), \quad k \in \mathbb{N}, \tag{7}$$

其中，$\bar{}\mathbf{u}$ 是已知的基准输入序列，通常选为前一时刻平移后的最优输入序列，即:

$$\bar{\mathbf{u}}(k) := \{u^*(1|k-1), \ldots, u^*(N-1|k-1), \bar{u}(N-1|k)\}, \tag{8}$$

其中，最后一个元素 $\bar{u}(N-1|k)$ 的常见取值为 r_u或 $u^*(N-1|k-1)$。在时刻 $k = T_{\text{ini}}$，当已有足够的输入–输出数据可供运行 GDPC 算法时，$\bar{}\mathbf{u}(k)$ 使用零输入序列（或基于经验的估计值）进行初始化。输入序列$\mathbf{u}_g$ 可通过求解二次规划，按下文所述在在线运行过程中自由优化。

使用一个分成两部分的持续激励输入序列，或两个不同的持续激励输入序列，我们可以如式（3）所示定义两个汉克尔数据矩阵，即:

$$\bar{H} := \begin{bmatrix} \bar{\mathbf{U}}_p \\ \bar{\mathbf{Y}}_p \\ \bar{\mathbf{U}}_f \\ \bar{\mathbf{Y}}_f \end{bmatrix} \in \mathbb{R}^{(n_u+n_y)(T_{\text{ini}}+N)\times T}, \qquad H := \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \\ \mathbf{Y}_f \end{bmatrix} \in \mathbb{R}^{(n_u+n_y)(T_{\text{ini}}+N)\times T_g}, \tag{9}$$

其中，第一部分/序列的长度 T 可以任意取大，第二部分/序列的长度 T_g 则可灵活选择。也就是说，T_g 应足够小，以满足计算要求，同时又应提供足够的自由度，以优化偏差/方差权衡。离线计算矩阵

$$\Theta := \bar{\mathbf{Y}}_f \begin{bmatrix} \bar{\mathbf{U}}_p \\ \bar{\mathbf{Y}}_p \\ \bar{\mathbf{U}}_f \end{bmatrix}^\dagger \in \mathbb{R}^{(n_y N)\times(T_{\text{ini}}(n_u+n_y)+n_u N)}. \tag{10}$$

在线时，在时刻 $k \geq T_{\text{ini}}$，给定 $\mathbf{u}_{\text{ini}}(k)$、$\mathbf{y}_{\text{ini}}(k)$ 和 $\bar{}\mathbf{u}(k)$，计算

$$\bar{\mathbf{y}}(k) = \Theta \begin{bmatrix} \mathbf{u}_{\text{ini}}(k) \\ \mathbf{y}_{\text{ini}}(k) \\ \bar{\mathbf{u}}(k) \end{bmatrix} \tag{11}$$

并求解*GDPC*优化问题:

$$\min_{\mathbf{g}(k),\mathbf{u}(k),\mathbf{y}(k),\sigma(k)} l_N(y(N|k)) + \sum_{i=0}^{N-1} l(y(i|k),u(i|k)) + \lambda_g l_g(\mathbf{g}(k)) + \lambda_\sigma l_\sigma(\sigma(k)) \tag{12a}$$

满足以下约束:

$$\begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \\ \mathbf{Y}_f \end{bmatrix} \mathbf{g}(k) = \begin{bmatrix} 0 \\ \sigma(k) \\ \mathbf{u}(k)-\bar{\mathbf{u}}(k) \\ \mathbf{y}(k)-\bar{\mathbf{y}}(k) \end{bmatrix}, \tag{12b}$$

$$(\mathbf{y}(k),\mathbf{u}(k)) \in \mathbb{Y}^N \times \mathbb{U}^N. \tag{12c}$$

上述代价函数 $l_N(y)$、$l(y,u)$、$l_g(g)$ 和 $l_\sigma(\sigma)$ 的定义方式与式（5）相同，其中 $Q, R \succ 0$。

由于矩阵 Θ 的大小不依赖于数据长度 T（即汉克尔矩阵 $\bar{}$ 的列数），因此，即使 T 很大，按照式（11）计算与 $\bar{}\mathbf{u}(k)$ 对应的预测输出也很高效。因此，GDPC 得益于无偏的基线输出预测，这使得所选的 T_g（即汉克尔矩阵 H 的列数）可以远小于 T。进而，这降低了 GDPC 的在线计算复杂度，且在含噪数据存在时不会牺牲性能。可以认为，对 T_g 的选择是在计算复杂度与可用于优化偏差-方差权衡的自由度之间取得折衷。

备注 III.1（无偏差 ***GDPC*** 设计）在实际应用中，实现无偏差跟踪是值得关注的问题。遵循 [21], 中针对 SPC 开发的无偏差设计，并将其进一步应用于 DeePC [13],，可以通过定义增量输入序列来设计无偏差 GDPC 算法

$$\Delta\mathbf{u}(k) := \Delta\bar{\mathbf{u}}(k) + \Delta\mathbf{u}_g(k), \quad k \in \mathbb{N},$$

其中 $\Delta\bar{}\mathbf{u}$ 选取为上一时刻平移后的最优输入序列，即

$$\Delta\bar{\mathbf{u}}(k) := \{\Delta u^*(1|k-1), \ldots, \Delta u^*(N-1|k-1), \Delta\bar{u}(N-1|k)\}. \tag{13}$$

$\bar{}$汉克尔矩阵 H 和 H 中的输入数据块必须$\bar{}$ $\bar{}$替换为增量输入数据，即 $\Delta\mathbf{U}_p$、$\Delta\mathbf{U}_f$ 和 $\Delta\mathbf{U}_p$、$\Delta\mathbf{U}_f$，分别如此。随后，施加到系统上的输入为 $u(k) := \Delta u^*(0|k) + u(k-1)$。

接下来，我们将对 GDPC 算法进行形式化分析。

A. GDPC 的适定性与设计

在本小节中，我们证明，在确定性情形下，GDPC 的预测轨迹是系统 (1) 的轨迹。在这种情形下，GDPC优化问题可简化为

$$\min_{\mathbf{g}(k),\mathbf{u}(k),\mathbf{y}(k)} l_N(y(N|k)) + \sum_{i=0}^{N-1} l(y(i|k),u(i|k)) \tag{14a}$$

满足约束：

$$\begin{bmatrix}\mathbf{U}_p\\ \mathbf{Y}_p\\ \mathbf{U}_f\\ \mathbf{Y}_f\end{bmatrix}\mathbf{g}(k) = \begin{bmatrix}0\\ 0\\ \mathbf{u}(k)-\bar{\mathbf{u}}(k)\\ \mathbf{y}(k)-\bar{\mathbf{y}}(k)\end{bmatrix}, \tag{14b}$$

$$(\mathbf{y}(k),\mathbf{u}(k)) \in \mathbb{Y}^N \times \mathbb{U}^N. \tag{14c}$$

引理 *III.2*（*GDPC* 适定性）考虑一个（或两个）长度足够的持续激励输入序列，并按 *(9)* 构造两个汉克尔矩阵，列数分别为 T 和 T_g，且 H 满行秩。再考虑由系统 *(1)* 生成的相应输出序列。对于任意给定的输入序列 $\bar{u}(k)$ 以及初始条件 $u_{ini}(k)$ 和 $y_{ini}(k)$，令 $\bar{y}(k)$ 按 *(11)* 定义。则存在一个实向量 $g(k) \in \mathbb{R}^{T_g}$，使得 *(14b)* 成立，当且仅当 $u(k)$ 和 $y(k)$ 是系统 *(1)* 的轨迹。

Proof: Define $\bar{\mathbf{g}}(k) := \begin{bmatrix}\bar{\mathbf{U}}_p\\ \bar{\mathbf{Y}}_p\\ \bar{\mathbf{U}}_f\end{bmatrix}^{\dagger}\begin{bmatrix}\mathbf{u}_{\text{ini}}(k)\\ \mathbf{y}_{\text{ini}}(k)\\ \bar{\mathbf{u}}(k)\end{bmatrix}$. Then it holds that:

$$\begin{aligned}\begin{bmatrix}\bar{H} & H\end{bmatrix}\begin{bmatrix}\bar{\mathbf{g}}(k)\\ \mathbf{g}(k)\end{bmatrix} &= \begin{bmatrix}\bar{\mathbf{U}}_p\\ \bar{\mathbf{Y}}_p\\ \bar{\mathbf{U}}_f\\ \bar{\mathbf{Y}}_f\end{bmatrix}\bar{\mathbf{g}}(k) + \begin{bmatrix}\mathbf{U}_p\\ \mathbf{Y}_p\\ \mathbf{U}_f\\ \mathbf{Y}_f\end{bmatrix}\mathbf{g}(k)\\ &= \begin{bmatrix}\mathbf{u}_{\text{ini}}(k)\\ \mathbf{y}_{\text{ini}}(k)\\ \bar{\mathbf{u}}(k)\\ \bar{\mathbf{y}}(k)\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \mathbf{u}(k)-\bar{\mathbf{u}}(k)\\ \mathbf{y}(k)-\bar{\mathbf{y}}(k)\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{u}_{\text{ini}}(k)\\ \mathbf{y}_{\text{ini}}(k)\\ \mathbf{u}(k)\\ \mathbf{y}(k)\end{bmatrix}.\end{aligned}$$

由于矩阵 H 满行秩，拼接矩阵 $\begin{bmatrix}\bar{H} & H\end{bmatrix}$ 也满行秩。因此，若使用一个输入序列，该结论由 [6] 得出；若使用两个不同的输入序列构造汉克尔矩阵，则由 [20] 得出。■

T_g 的选择使计算复杂度与可用于改进由已知基准输入序列生成的输出序列的自由度之间得以权衡。事实上，较大的 T_g 会增大数据矩阵 $\begin{bmatrix}\mathbf{U}_p\end{bmatrix}\mathbf{Y}$ 的零空间，从而在确定性情形下限制 $\mathbf{g}(k)$。然而，高性能也可以在含噪数据情形下通过较小的 T_g 实现，因为基准预测输出是使用无偏最小二乘预测器计算的，即按式 (11) 所定义的方式计算。

定义已知输入序列$\bar{\mathbf{u}}(k)$的另一种方法是使用无约束SPC控制律 [5]。为此，注意到矩阵 Θ 可以划分为如下形式（例如，参见[21], ）：$\begin{bmatrix}P_1 & P_2 & \Gamma\end{bmatrix}$

$$\bar{\mathbf{y}}(k) = \begin{bmatrix}P_1 & P_2\end{bmatrix}\begin{bmatrix}\mathbf{u}_{\text{ini}}(k)\\ \mathbf{y}_{\text{ini}}(k)\end{bmatrix} + \Gamma\bar{\mathbf{u}}(k).$$

然后，定义 $\Psi := \text{diag}\{R,\dots,R\}$、$\Omega := \text{diag}\{Q,\dots,Q,\alpha Q\}$、$G := 2(\Psi + \Gamma^T\Omega\Gamma)$ 以及 $F: = 2\Gamma^T\Omega$，可得：

$$\begin{aligned}&\bar{\mathbf{u}}_{\text{spc}}(k)\\ &:= -G^{-1}\left(F\left(\begin{bmatrix}P_1 & P_2\end{bmatrix}\begin{bmatrix}\mathbf{u}_{\text{ini}}(k)\\ \mathbf{y}_{\text{ini}}(k)\end{bmatrix} - \mathbf{r}_y\right) - 2\Psi\mathbf{r}_u\right),\end{aligned} \tag{15}$$

其中，$\mathbf{r}_y := \text{col}(r_y,\dots,r_y)$，且 $\mathbf{r}_u := \text{col}(r_u,\dots,r_u)$。由于 G 的逆矩阵是在离线阶段计算的，因此即使数据长度 T很大，在线计算$\bar{\mathbf{u}}_{\text{spc}}(k)$在数值上仍然高效。在这种情况下，由于相应的基准预测输出轨迹无偏，因此在式（12a）中可以采用更简单的正则化代价 $l_g(\mathbf{g}) := \|\mathbf{g}\|_2^2$，而不损失一致性。当基准输入序列按式（15）计算时，优化输入序列 $\mathbf{u}_g(k)$可在无约束SPC轨迹违反约束时用于强制满足约束；在适当调节 λ_g的情况下，它还可以优化偏差与方差之间的权衡。

B. GDPC 的稳定性

本节将给出 GDPC 具有渐近稳定性的充分条件。为此，定义 $J(\mathbf{y}(k),\mathbf{u}(k)) := l_N(y(N|k)) + \sum_{i=0}^{N-1} l(y(i|k)\ u(i|k))$，令$l_N(y) := l(y,0)$，并令 $\mathbf{y}^*(k)$ 和 $\mathbf{u}^*(k)$ 表示时刻 $k \geq T_{\text{ini}}$ 的最优轨迹。给定时刻 $k \geq T_{\text{ini}}$ 的最优输入序列，即 $\mathbf{u}^*(k) = \bar{\mathbf{u}}(k) + \mathbf{u}_g^*(k)$，定义时刻 $k+1$ 的次优输入序列如下：

$$\begin{aligned}\mathbf{u}_s(k+1) &= \bar{\mathbf{u}}(k+1) + \mathbf{u}_g(k+1)\\ &= \text{col}(u^*(1|k),\dots,u^*(N-1|k),\bar{u}(N|k)) \quad (16)\\ &\quad + \text{col}(0,\dots,0,0), \quad (17)\end{aligned}$$

并令

$$\begin{aligned}\mathbf{y}_s(k+1) &= \bar{y}(k+1)\\ &= \text{col}(y^*(2|k),\dots,y^*(N|k),\bar{y}(N+1|k))\end{aligned}$$

表示相应的次优输出轨迹。注意，次优输出序列中的最后一个输出满足：

$$\bar{y}(N+1|k) = \sum_{i=1}^{T_{\text{ini}}} a_i y^*(N+1-i|k) + \sum_{i=1}^{T_{\text{ini}}} b_i u^*(N+1-i|k).$$

定义 III.3（类 $\mathcal{K}$ **函数**）函数 $\varphi: \mathbb{R}_+ \to \mathbb{R}_+$ 属于类 $\mathcal{K}$，当且仅当它连续且严格递增

且 $\varphi(0) = 0$。函数 $\varphi: \mathbb{R}_+ \to \mathbb{R}_+$ 属于类$\mathcal{K}_\infty$，当 $\varphi \in \mathcal{K}$ 且 $\lim_{s\to\infty} \varphi(s) = \infty$ 时。id 表示恒等$\mathcal{K}_\infty$ 函数，即 $\mathrm{id}(s) = s$。

接下来，如 [11], 所提出的，我们定义一个非最小状态:

$$\mathbf{x}_{\text{ini}}(k) := \\ \mathrm{col}(y(k-T_{\text{ini}}), \dots, y(k-1), u(k-T_{\text{ini}}), \dots, u(k-1)),$$

以及函数 $W(\mathbf{x}_{\text{ini}}(k)) := \sum_{i=1}^{T_{\text{ini}}} l(y(k-i)\ u(k-i))$。为便于阐述，下面假设 $r_y = 0$ 和 $r_u = 0$。然而，对于任何与可行稳态相容的常值参考值，同样的证明也适用。

假设 III.4（终端稳定条件）对于任意可行初始状态 $\mathbf{x}_{\text{ini}}(k)$，都存在一个函数$\rho \in \mathcal{K}_\infty$，其 ρ <为恒等映射、预测时域为 $N \geq T_{\text{ini}}$，且$\bar{u}(N|k) \in \mathbb{U}$，使得 $\bar{y}(N+1|k) \in \mathbb{Y}$ 且

$$l(\bar{y}(N+1|k), \bar{u}(N|k)) - (\mathrm{id} - \rho) \circ l(y(k-T_{\text{ini}}), u(k-T_{\text{ini}})) \leq 0. \tag{18}$$

定理 III.5（***GDPC*** 的稳定性）假设存在 $\alpha_{1,\ l}, \alpha_{2,\ l}, \alpha_{2,\ J} \in \mathcal{K}_\infty$，使得 $\forall\ (y, u)\ \in \mathbb{Y} \times \mathbb{U}$

$$\alpha_{1,l}(\|\mathrm{col}(y,u)\|) \leq l(y,u) \leq \alpha_{2,l}(\|\mathrm{col}(y,u)\|), \tag{19a}$$

$$J(\mathbf{y}^*(k), \mathbf{u}^*(k)) \leq \alpha_{2,J}(\|\mathbf{x}_{ini}(k)\|), \quad \forall \mathbf{x}_{ini}(k) \in \mathcal{N}, \tag{19b}$$

对于某个以原点为内部点的适当集合 $\mathcal{N}$。此外，假设假设 ***III.4*** 成立，并且问题(14) 对所有 $k \geq T_{ini}$均可行。那么，与求解问题(14) 的 *GDPC* 算法构成闭环的系统 (1) 渐近稳定。

证明: 与 [11] 中的 DeePC 算法类似，我们考虑以下储能函数

$$V(\mathbf{x}_{\text{ini}}(k)) := J(\mathbf{u}^*(k), \mathbf{y}^*(k)) + W(\mathbf{x}_{\text{ini}}(k)),$$

并证明它是正定的，且满足耗散不等式。首先，由 (19a) 以及 [11] 中的引理14可知，存在 $\alpha_{1,V}$,$\alpha_{2,V} \in \mathcal{K}_\infty$，使得

$$\alpha_{1,V}(\|\mathbf{x}_{\text{ini}}(k)\|) \leq V(\mathbf{x}_{\text{ini}}(k)) \leq \alpha_{2,V}(\|\mathbf{x}_{\text{ini}}(k)\|).$$

然后，定义供给函数

$$s(y(k-T_{\text{ini}}), u(k-T_{\text{ini}})) := \\ l(\bar{y}(N+1|k), \bar{u}(N|k)) - l(y(k-T_{\text{ini}}), u(k-T_{\text{ini}})).$$

根据最优性原理，有

$$\begin{aligned} &V(\mathbf{x}_{\text{ini}}(k+1)) - V(\mathbf{x}_{\text{ini}}(k)) = \\ &\quad J(\mathbf{y}^*(k+1), \mathbf{u}^*(k+1)) + W(\mathbf{x}_{\text{ini}}(k+1)) \\ &\quad - J(\mathbf{y}^*(k), \mathbf{u}^*(k)) - W(\mathbf{x}_{\text{ini}}(k)) \\ &\quad \leq J(\mathbf{y}_s(k+1), \mathbf{u}_s(k+1)) - J(\mathbf{y}^*(k), \mathbf{u}^*(k)) \\ &\quad + W(\mathbf{x}_{\text{ini}}(k+1)) - W(\mathbf{x}_{\text{ini}}(k)) \\ &\quad = l(\bar{y}(N+1|k), \bar{u}(N|k)) - l(y(0|k), u(0|k)) \\ &\quad + l(y(0|k), u(0|k)) - l(y(k-T_{\text{ini}}), u(k-T_{\text{ini}})) \\ &\quad = s(y(k-T_{\text{ini}}), u(k-T_{\text{ini}})). \end{aligned}$$

因此，储能函数 $V(\mathbf{x}_{\text{ini}}(k))$ 沿闭环轨迹满足耗散不等式。由于根据式 (18)，供给函数满足

$$s(y(k-T_{\text{ini}}), u(k-T_{\text{ini}})) \leq -\rho \circ l(y(k-T_{\text{ini}}), u(k-T_{\text{ini}})),$$

由此即可根据 [11] 中的推论17 得到所述结论。注意，■ 条件 (18) 是 [11], 中推论17所采用的条件 (25) 的一个特例，即对于 $M = 1$。

注释 III.6（终端稳定条件） 终端稳定条件 (18) 可视为一个隐式条件，即通过选择足够大的预测时域 N，该条件更有可能成立。或者，也可以将其作为显式约束加入问题 (14)：在预测输入和输出序列的末尾分别再添加一个输入和输出$\mathbf{u}(k)$、$\mathbf{y}(k)$。这还需要在相应的 H 和 H 汉克尔矩阵中纳入所需的额外数据¯。由此得到一个凸二次约束二次规划，仍可高效求解。稳定条件 (18) 也可用于正则化数据驱动预测控制问题 (12)，以确保收敛；但在这种情况下，建议采用软约束实现，以防因含噪数据而导致问题不可行。

值得指出的是， [9]中采用的条件意味着条件 (18) 成立，即假设 III.4 的保守性较低。事实上，若在问题 (14) 中施加终端等式约束，即 $y(N|k) = r_y$，则在时刻 $k+1$，$\bar{u}(N|k) = r_u$ 是一个可行选择，由此得到 $\bar{y}(N+1|k) = r_y$，进而有 $l(\bar{y}(N+1|k)\ \bar{u}(N|k)) = 0$。因此，由于阶段代价 $l(y,u)$ 是正定的，且 ρ <id，条件 (18) 显然成立。或者，也可以采用数据驱动方法，在$\mathbf{x}_{\text{ini}}$空间中计算一个合适的不变终端集，如 [22] 中所述。目前可通过线性矩阵不等式，以可处理的方式基于数据计算椭球集形式的不变集；该方法还会得到一个需要在线求解的凸二次约束二次规划。

IV. 仿真结果

在本节中，我们考虑一个基于波音747纵向运动飞行控制的基准模型预测控制示例，参见 [4, 第 6 节。4] 对 $T_s = 0.1[s]$ 采用零阶保持进行离散化后，我们得到如下式（1）所示的离散时间线性模型:

$$\begin{aligned} A &= \begin{bmatrix} 0.9997 & 0.0038 & -0.0001 & -0.0322 \\ -0.0056 & 0.9648 & 0.7446 & 0.0001 \\ 0.0020 & -0.0097 & 0.9543 & -0.0000 \\ 0.0001 & -0.0005 & 0.0978 & 1.0000 \end{bmatrix}, \\ B &= \begin{bmatrix} 0.0010 & 0.1000 \\ -0.0615 & 0.0183 \\ -0.1133 & 0.0586 \\ -0.0057 & 0.0029 \end{bmatrix}, \\ C &= \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & -1.0000 & 0 & 7.7400 \end{bmatrix}, \end{aligned} \tag{20}$$

其中包含两个输入，分别为油门 u_1 和 u_2升降舵角度，以及两个输出，分别为纵向速度和爬升率。输入和输出受到如下约束：

$$\mathbb{U} := \left\{ u \in \mathbb{R}^2 \ : \ \begin{bmatrix} -20 \\ -20 \end{bmatrix} \leq u \leq \begin{bmatrix} 20 \\ 20 \end{bmatrix} \right\}$$
$$\mathbb{Y} := \left\{ y \in \mathbb{R}^2 \ : \ \begin{bmatrix} -25 \\ -15 \end{bmatrix} \leq y \leq \begin{bmatrix} 25 \\ 15 \end{bmatrix} \right\}.$$

预测控制器的代价函数使用 $\lambda_g = 10^5$、$\lambda_\sigma = 10^7$、$N = 20$、$T_{ini} = 20$、$Q = 10 \cdot I_{n_y}$ 和 $R = 0.01 \cdot I_{n_u}$，按照式（4）和式（5）定义。对于 GDPC，如果按照式（8）计算次优输入序列，并使用上一时刻平移后的最优序列，且满足 $\bar{u}(k+N-1) = u^*(k-2+N)$，则正则化代价 $l_g(\mathbf{g}(k))$ 按照式（6）定义。如果使用无约束 SPC 解计算次优输入序列，则使用正则化代价 $l_g(\mathbf{g}(k)) = \lambda_g \|\mathbf{g}(k)\|_2^2$。对于 DeePC，正则化代价 $l_g(\mathbf{g}(k))$ 按照式（6）定义。这样，所有 3 种用于比较的预测控制算法都使用一致的输出预测器。实时二次规划（或带二次约束的二次规划）控制问题使用 Mosek [23] 求解，运行设备是一台搭载 Intel i7-9750H CPU 和 16GB 内存的笔记本电脑。

下文将仿真结果分为 3 个小节，分别讨论标称无噪数据下的性能、低噪声方差和高噪声方差下含噪数据的性能，以及与 DeePC 的比较。收集到 T_{ini} 个样本后，控制器才会启动。在截至 T_{ini} 的各个时刻，系统由一个较小的随机输入驱动。为确保比较合理，所有仿真/预测控制器在截至 T_{ini} 时使用的随机输入信号都相同。在数据生成实验中，输入序列由介于 $[-3, 3]$ 之间的伪随机二进制序列信号构成。

A. 无噪数据下的 *GDPC* 性能

在本次仿真中，我们采用式 (8) 计算得到的次优输入序列来实现 GDPC 算法。图 2 展示了输出、输入以及优化后的输入 $\mathbf{u}_g$ 随时间的变化。

我们观察到，GDPC 闭环轨迹收敛至参考值，且优化后的输入仅在开始时起作用；此后，移位后的次优序列便成为最优序列。沿着这些轨迹，稳定条件 (18) 得到隐式满足；若在线施加该条件，GDPC 问题将递归可行，并产生相同的轨迹。

B. 含噪数据下的 *GDPC* 性能

接下来，我们展示 GDPC 在低方差和高方差含噪数据下的性能。

图 3 展示了使用 $\bar{u}(k)$ 的 GDPC 响应，其中 $\bar{u}(k)$ 如式 (15) 所定义，即采用无约束 SPC 解。尽管计算基准输入序列 $\bar{u}(k)$ 的两种不同方法所得的总输入 $u(k)$ 差异很小，但输入的优化部分 $u_g(k)$ 却有显著差异。对于采用无约束 SPC 的 GDPC 算法，我们可以看到，优化输入仅

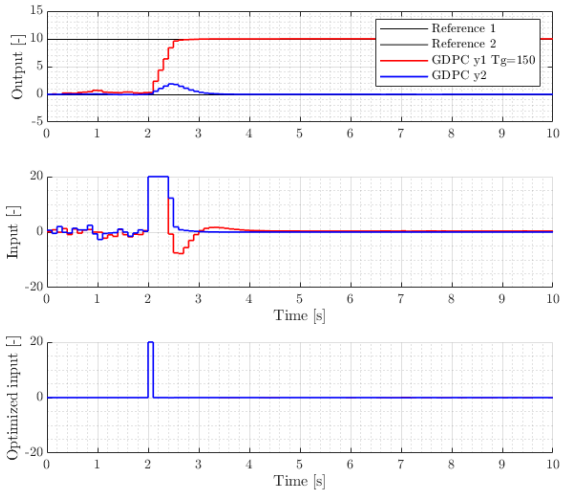


图 1. GDPC 跟踪性能： $T_g = 150$, $T = 1000$, $\sigma_w^2 I = 0 I$。

.

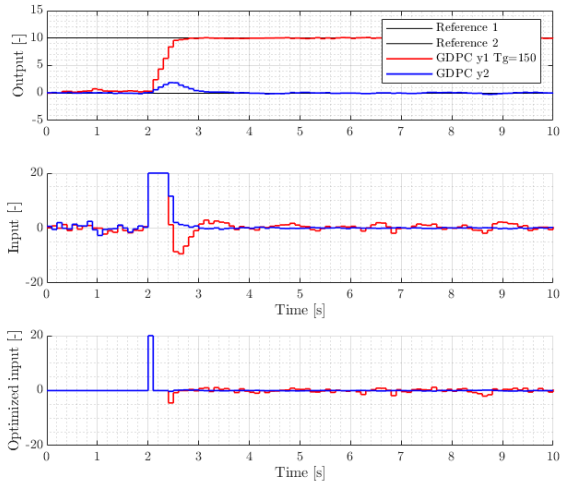


图 2. GDPC 跟踪性能： $T_g = 150$, $T = 1000$, $\sigma_w^2 I = 0.05 I$,

$\bar{u}$ 8 ，如（）所示。

用于施加约束，而在稳态附近，无约束 SPC 则成为最优解。

接下来，我们展示 GDPC 在高方差噪声下的性能；这种情况也需要适当增加数据规模。图 5 展示了使用无约束 SPC 解计算的 $\bar{\mathbf{u}}(k)$ 对应的 GDPC 仿真结果。在这种情况下，优化后的输入在稳态附近也处于激活状态，这表明 GDPC 确实针对 SPC 解优化了偏差-方差权衡。图 6 显示，DeePC 需要长度为 750 的数据序列，才能达到与数据长度为 $T_g = 250$ 的 GDPC 相近的性能（这与在线复杂度相关），$T = 5000$。在高方差噪声仿真中，三个受测预测控制器的平均计算时间如下：采用式 (8) 中 $\bar{u}$ 的 GDPC：70 毫秒；

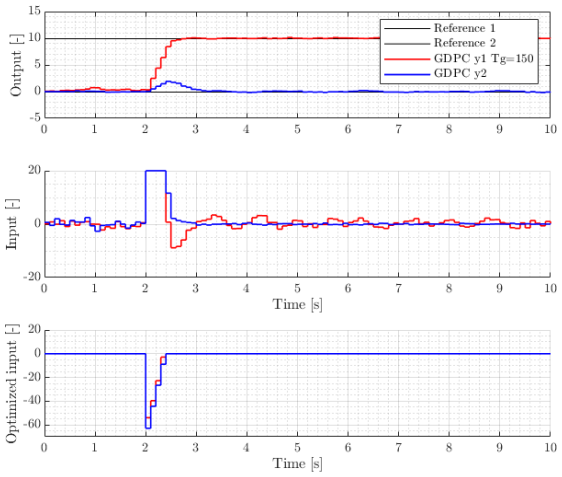


图 3。GDPC 跟踪性能： $T_g = 150$、 $T = 1000$、 $\sigma_w^2 I = 0.05I$、 $\bar{u}$ 15 ，如（）所示。

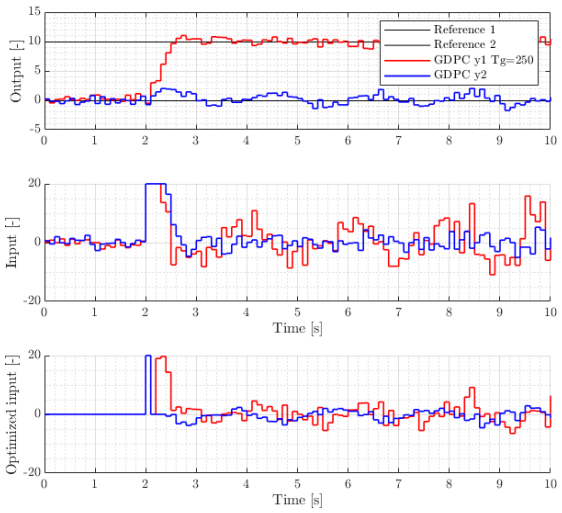


图 4。GDPC 跟踪性能： $T_g = 250$、 $T = 5000$、 $\sigma_w^2 I = 0.5I$、 $\bar{u}$ 8 ，如（）所示。

采用式 (15) 中 $\bar{u}$ 的 GDPC：30 毫秒；DeePC：680 毫秒。这表明，对于高噪声和大规模系统，采用式 (15) 中 $\bar{u}$ 的 GDPC 是效率最高的方案。

C. 多次运行下与 *DeePC* 的比较

在本节中，我们比较 GDPC 与 DeePC 在不同大小的 T 下多次运行的性能。性能可表示为 [14]:

$$\mathcal{J} = \sum_{k=T_{\text{ini}}}^{t_{\max}} \|Q^{\frac{1}{2}}(y(k) - r_y(k))\|_2^2 + \|R^{\frac{1}{2}}(u(k) - r_u(k))\|_2^2 \tag{21}$$

$$\mathcal{J}_u = \sum_{k=T_{\text{ini}}}^{t_{\max}} \|u(k)\|_2^2, \tag{22}$$

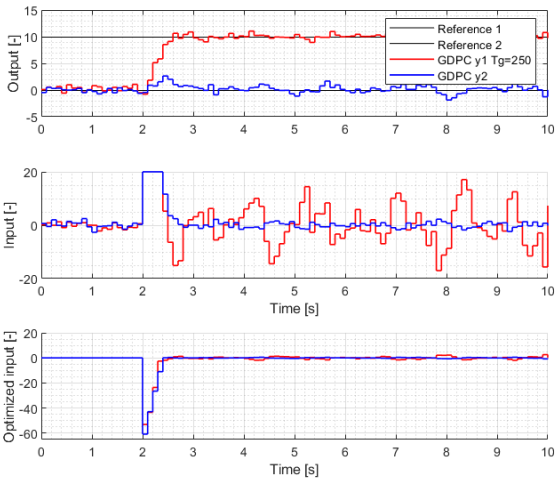


图 5。GDPC 跟踪性能： $T_g = 250$、 $T = 5000$、 $\sigma_w^2 I = 0.5I$、 $\bar{u}$ 15 ，如（）所示。

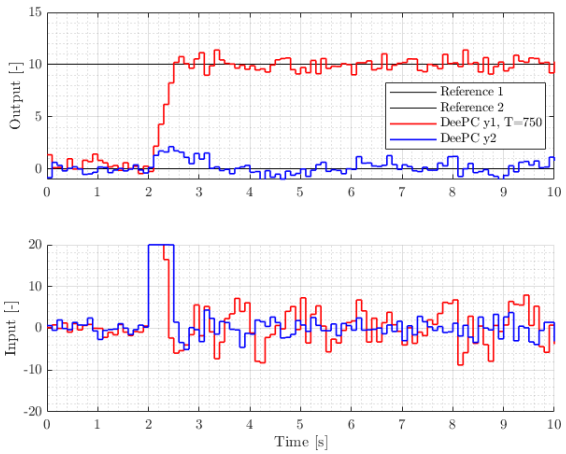


图 6。DeePC 跟踪性能： $T = 750$、 $\sigma_w^2 I = 0.5I$。

其中， $t_{\max}$ 为仿真时间。请注意，性能得分是在仿真结束后计算的，因此使用的是仿真数据（而非预测数据）。此外，数据采集实验和仿真都会受到方差为 $\sigma_w^2 I = 0.05I$的噪声影响。

如图 7 所示，我们注意到，数据长度为 $T_g = 150$ 的 GDPC 预测器（基于汉克尔矩阵的预测器）和数据长度为 $T = 1000$ 的 GDPC 预测器（基于最小二乘的预测器）能够达到数据长度为 $T = 500$ 的 DeePC 的性能，而数据长度为 $T = 150$ 时的 DeePC 性能较低。此外，与数据长度为 $T = 500$ 的 DeePC 的相同成本相比，GDPC 的输入成本 $\mathcal{J}_u$ 更低。

	GDPC $T_g = 150$	DeePC $T = 150$	DeePC $T = 500$
t_{cpu}	33ms	31ms	235毫秒

从计算复杂度的角度来看，如表1所示，GDPC 在 $T_g = 150$下的平均 CPU 时间

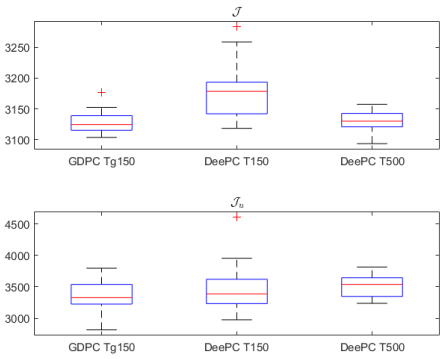


图 7. GDPC 与 DeePC 的性能比较，$\sigma_w^2 I = 0.05I$。

与 DeePC 在 $T = 150$下的平均 CPU 时间处于同一数量级，而 DeePC 在 $T = 500$ 下的平均 CPU 时间约为其 8 倍。

所得结果验证了 GDPC 在含噪数据下的控制性能与在线计算复杂度之间提供了更灵活的权衡。因此，GDPC 为工程师提供了一种实用且鲁棒的数据驱动预测控制器，适用于实时实现。

V. 结论

本文开发了一种广义数据驱动预测控制器，该控制器将预测输入序列构造为已知次优输入序列与在线计算得到的优化输入序列之和。这使我们能够结合两种数据驱动预测器：一种基于最小二乘的无偏预测器，用于计算次优输出轨迹；另一种基于汉克尔矩阵的数据使能预测器，用于计算优化输出轨迹。我们证明，这种公式化方法可得到适定的数据驱动预测控制器，其稳定性性质与 DeePC 相似。此外，通过仿真，我们表明，对于汉克尔矩阵预测器而言，所开发的 GDPC 算法在数据序列较短时，也能达到 DeePC 在数据序列较长时的性能，因此在计算上更具优势。

参考文献

[1] Z.-S. Hou 和 Z. Wang，《从基于模型的控制到数据驱动控制：综述、分类与展望》，*Information Sciences*，第235卷，第3–35页，2013年。[2] F. Lamnabhi-Lagarrigue、A. Annaswamy、S. Engell、A. Isaksson、P. Khargonekar、R. M. Murray、H. Nijmeijer、T. Samad、D. Tilbury 和 P. Van den Hof，《面向人类未来的系统与控制：研究议程——当前与未来的作用、影响及重大挑战》，*Annual Reviews in Control*，第43卷，第1–64页，2017年。[3] J. M. Maciejowski，《*Predictive Control with Constraints*》。Prentice Hall，2002年。[4] E. F. Ca macho 和 C. Bordons，《模型预测控制》。Springer Verlag，2007年。[5] W. Favoreel、B. De Moor 和 M. Gevers，《SPC：子空间预测控制》，载于*IFAC 第14届世界大会论文集*，第32卷，第2期。Elsevier，1999年，第4004–4009页。[6] J. Coulson、J. Lygeros 和 F. D¨orfler，《数据赋能预测控制：浅谈 DeePC》，载于第*18*届欧洲控制会议，意大利那不勒斯，2019年，第307–312页。

[7] J. C. Willems、P. Rapisarda、I. Markovsky 和 B. L. De Moor，《关于持续激励的简短说明》，《系统与控制快报》，第54卷，第4期，第325–329页，2005年。[8] H. Yang 和 S. Li，《基于降阶汉克尔矩阵的数据驱动预测控制器设计》，载于 IEEE《第10届亚洲控制会议（ASCC）论文集》。马来西亚亚庇：IEEE，2015年，第1–7页。[9] J. Berberich、J. K¨ohler、M. A. M¨uller 和 F. Allg¨ower，《具有稳定性与鲁棒性保证的数据驱动模型预测控制》，《IEEE自动控制汇刊》，第66卷，第4期，第1702–1717页，2021年。[10] F. Fiedler 和 S. Lucia，《数据赋能预测控制与子空间预测控制之间的关系》，载于 IEEE《欧洲控制会议（ECC）论文集》，荷兰鹿特丹，2021年，第222–229页。[11] M. Lazar，《用于分析预测控制器稳定性的基于耗散性的框架》，载于 IFAC PapersOnLine 54–6，第七届 IFAC非线性模型预测控制会议，斯洛伐克布拉迪斯拉发，2021年，第159–165页。[12] F. Dorfler、J. Coulson 和 I. Markovsky，《通过正则化与松弛衔接直接和间接数据驱动控制表述》，《IEEE自动控制汇刊》，第1–1页，2022年。[13] M. Lazar 和 P. C. N. Verheijen，《无偏差数据驱动预测控制》，载于2022年 IEEE 第61届决策与控制会议（CDC），2022年，第1099–1104页。[14] V. Breschi、A. Chiuso 和 S. Formentin，《随机环境下的数据驱动预测控制：统一框架》，《自动化》，第152卷，文章编号110961，2023年。[15] K. Zhang、Y. Zheng 和 Z. Li，《面向高效数据赋能预测控制的降维》，arXiv，2022年。[16] S. Baros、C.-Y. Chang、G. E. Col´on-Reyes 和 A. Bernstein，《在线数据赋能预测控制》，《自动化》，第138卷，109926，2022年。[17] E. Elokda、J. Coulson、P. N. Beuchat、J. Lygeros 和 F. D¨orfler，《面向四旋翼无人机的数据赋能预测控制》，《国际鲁棒与非线性控制杂志》，第31卷，第8916–8936页，2021年。[18] J. Berberich、J. K¨ohler、M. A. M¨uller 和 F. Allg¨ower，《数据驱动模型预测控制：闭环保证与实验结果》，《at—自动化技术》，第69卷，618，2021 7，608号，第–页。[19] P. G. Carlet、A. Favato、S. Bolognani 和 F. D¨orfler，《面向同步电机驱动系统的数据驱动连续集合预测电流控制》，《IEEE电力电子汇刊》，第37卷，第6期，第6637–6646页，2022年。[20] H. J. van Waarde、C. De Persis、M. K. Camlibel 和 P. Tesi，《状态空间系统的 Willems基本引理及其向多个数据集的扩展》，《IEEE控制系统快报》，第4卷，第3期，第602–607页，2020年。[21] P. C. N. Verheijen、G. R. Gonc.alves da Silva 和 M. Lazar，《具有估计预测矩阵的数据驱动速率积分预测控制》，载于 IEEE《第25届国际系统理论、控制与计算会议（ICSTCC）论文集》，罗马尼亚锡纳亚，2021年，第630–636页。[22] J. Berberich、J. K¨ohler、M. A. M¨uller 和 F. Allg¨ower，《数据驱动模型预测控制终端要素的设计》，IFAC–PapersOnLine，第54卷，第6期，第257–263页，2021年，第七届 IFAC 非线性模型预测控制会议 NMPC 2021。[23] M. ApS，《MOSEK MATLAB 优化工具箱手册》。9.0版，2019年。[在线]。网址：docs.mosek.com