2025 American Control Conference (ACC)
July 8-10, 2025. Denver, CO, USA

Data-Driven Minimum-Gain Pole Placement

Ananta Kant Rai and Vaibhav Katewa

Abstract— **Minimum-gain pole placement is a classical problem that aims to find a static state feedback matrix with the minimum norm that places the closed-loop poles at desired locations. In this paper, we present the direct data-driven formulation of this problem without identifying the system model. We derive and discuss the conditions for pole placement using data matrices, and propose a projection-based gradient descent algorithm to solve the problem. We also consider sparsity constraints on the feedback matrix and obtain approximately sparse solutions. Our simulations show that the proposed direct method is more accurate than the model-based approach in placing the poles as well as in obtaining a feedback matrix with lower norm.**

I. INTRODUCTION

The pole placement problem for linear time-invariant (LTI) systems is a classical problem in the control theory. It involves designing a static feedback controller to place the poles of a closed-loop system at some desired locations to achieve certain desired behaviors such as stability, robustness, etc. The feedback gain matrix determined as the solution to the pole placement problem is not unique in general, especially for multi-input multi-output (MIMO) systems. The non-uniqueness provides a choice to select a feedback matrix that ensures certain other control objectives in addition to pole placement like minimum-norm gain, robustness, enforcing certain sparsity structure on the feedback matrix, etc.

Traditionally, the pole placement problem is model-based and non-sparse. It relies highly on the precise knowledge of system matrices to determine the feedback gain matrix. However, obtaining accurate models can be challenging due to uncertainties, nonlinearities, or incomplete system identification. This motivates us to leverage the data-driven control methods to address the pole placement problem. These methods utilize data generated by the system to achieve the design objectives, where system identification may or may not be required.

In this paper, we address the non-sparse and sparse minimum-gain pole placement problems. We use the open-loop data of the system and propose algorithms to design the feedback gain matrix with minimum-norm and desired sparsity structure.

Related Work: The model-based, non-sparse pole placement problem is well studied in literature [1], [2], [3]. Building on these, problems to achieve certain design and control objectives like robustness [4], [5] and minimum-gain feedback [6], [7], [8] were addressed. In recent times, the sparse feedback problem has gained attention as sparsity patterns appear inherently in interconnected systems and decentralized control problems [9]. In [8], iterative algorithms were provided for minimum-gain approximately sparse and sparse feedback design problems, and it was shown that the sparse problem is NP-hard.

While these works addressed various types of pole placement problems, they assume the availability of highly reliable system models. This may pose a challenge when system identification is infeasible and inaccurate [10]. In recent times, data-driven methods have gained attention for various control problems [11], [12]. The data-driven pole placement problem is addressed in [13], [14]. In [13], the exact pole placement and eigenstructure assignment problem is studied in contrast to [15], where robustness guarantees are provided for pole placement in linear matrix inequality (LMI) regions. In contrast, we study the minimum-gain pole placement problem which is more difficult than the pole placement problem. Also, in contrast to [13], we present an alternate formulation for data-driven pole placement that relies on Sylvester equation-based parameterization [3]. The data-driven sparse feedback problem is recently studied in [14] in which the set of all feasible gain matrices is characterized using the open-loop data. Using this, the authors also present minimum-gain pole placement formulation. However, this formulation and the corresponding constraints are at a more abstract level, which might be difficult to implement and solve. In contrast, we address the minimum-gain pole placement more explicitly by computing the gradient and then using the projection-based gradient-descent to arrive at the solution.

The main contributions of this paper are:

1. We present data-driven formulation of the non-sparse and approximately-sparse minimum-gain pole placement problems.
2. We provide a projection-based gradient-descent algorithm to numerically solve the above problems.
3. We present numerical simulations to compare the model-based and direct methods in the presence of noisy data. Our results show that direct methods perform better than model-based methods when data is corrupted with noise.

Mathematical Notations: $\|\cdot\|$ denotes the Frobenius norm of a matrix throughout this paper. $(\cdot)^T, (\cdot)^*$ and $(\cdot)^H$ denote the transpose, conjugate and conjugate transpose of a vector/matrix. $\text{Tr}(\cdot)$ denotes the trace of a matrix. $\circ$ and $\otimes$ denote the Hadamard product and Kronecker product, respectively. $(\cdot)^\dagger$ denotes the Moore-Penrose inverse of a matrix. I denotes

Ananta Kant Rai is with the Department of Electrical Communication Engineering (ECE) at the Indian Institute of Science (IISc), Bangalore. Vaibhav Katewa is with the Robert Bosch Center for Cyber-Physical Systems and the Department of ECE at IISc Bangalore. Email IDs: anantakant@iisc.ac.in, vkatewa@iisc.ac.in

This work is supported in part by SERB grant MTR/2022/000522.

the identity matrix. vec$(\cdot)$ and Mat$(\cdot)$ denote the vectorization of a matrix and reverse of vectorization of a matrix, respectively. Diag$(\cdot)$ denotes the diagonalization of a vector. $j = \sqrt{-1}$ denotes the unit imaginary number. $(\cdot)^R$ and and $(\cdot)^I$ denote the real and imaginary parts of a complex entity, respectively. $\mathbf{1}_n$ and $\mathbf{1}_{m\times n}$ denote a vector and matrix of all ones, respectively. Null$(\cdot)$, Range$(\cdot)$ and Rank$(\cdot)$ denotes the null space, range space and rank of a matrix, respectively.

II. Problem Formulation

We consider the following discrete linear time-invariant (LTI) system

$$x(t+1) = Ax(t) + Bu(t), \tag{1}$$

where $t \in \mathbb{R}_{\geq 0}$ denotes the time instant, $x \in \mathbb{R}^n, u \in \mathbb{R}^m$ are the state and input vectors, $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$ are the system and input matrices, respectively. We assume a static state feedback controller of the form $u(t) = -Kx(t)$, where $K \in \mathbb{R}^{m\times n}$ is the feedback matrix. The resulting closed-loop dynamical system is given by

$$x(t+1) = (A - BK)x(t). \tag{2}$$

The feedback modifies the dynamics of the open-loop system (1) and affects its properties and performance. One of the problems of interest is to select K such that the eigenvalues/poles of the closed-loop matrix $A-BK$ coincide with a given set of eigenvalues/poles, which is known as the Pole Placement (PP) problem. By assigning eigenvalues at appropriate locations, one can achieve certain desired properties like stability, response time etc.

Let $\mathcal{P} = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ denote the set of desired eigenvalues of the closed-loop system. Since $A - BK$ is real, we assume that the set $\mathcal{P}$ is closed under complex conjugation. Define $\Lambda = \text{Diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ as the diagonal matrix consisting of the desired eigenvalues. Further, let $X \in \mathbb{C}^{n\times n}$ be an invertible eigenvector matrix whose columns are the eigenvectors of $A - BK$. Then, the Pole Placement (PP) problem is

$$\textbf{PP: Find } (K, X) \text{ s.t. } (A-BK)X = X\Lambda, \text{Rank}(X) = n. \tag{3}$$

Since the uncontrollable modes of (A, B) cannot be shifted by static state feedback K, we make the following assumption.

Assumption 1. *For the pole placement problem, all uncontrollable modes of (A, B) are contained in $\mathcal{P}$.*

For MIMO systems, the solutions (K, X) to (3) are not unique in general. Thus, there is flexibility to choose K to achieve certain design objectives. In this paper, we focus on determining the minimum-norm gain matrix K that achieves the desired pole placement. This is called the Minimum-Gain Pole Placement (MGPP) problem and is formulated as

$$\textbf{MGPP}: \min_{K\in\mathbb{R}^{m\times n}, X\in\mathbb{C}^{n\times n}} \frac{1}{2}\|K\|^2 \tag{4}$$
$$\text{s.t.} \quad (A - BK)X = X\Lambda \tag{4a}$$

Often, there may be some sparsity constraints present on the entries of K, that is, only certain entries of K are allowed to be non-zero. This arises especially in multi-agent networked settings where an agent may not have access to the states of all other agents. To address this, we impose sparsity constraints on K as follows. Let $S \in \{0, 1\}^{m\times n}$ be a binary sparsity matrix specifying the sparsity pattern of K as

$$K_{ij} = \begin{cases} * & \text{if} \quad S_{ij} = 1, \\ 0 & \text{if} \quad S_{ij} = 0, \end{cases}$$

where $* \in \mathbb{R}$ is a scalar. Then, the sparsity constraints are specified as

$$S^c \circ K = 0, \tag{5}$$

where $S^c \triangleq 1_{m\times n} - S$ is complement of the sparsity matrix. The case where $S = 1_{m\times n}$ is the non-sparse case. With this, we formulate the MGPP problem with sparsity constraints as

$$\textbf{Sparse MGPP}: \min_{K\in\mathbb{R}^{m\times n}, X\in\mathbb{C}^{n\times n}} \frac{1}{2}\|K\|^2 \tag{6}$$
$$\text{s.t.} \quad (A - BK)X = X\Lambda \tag{6a}$$
$$S^c \circ K = 0. \tag{6b}$$

Due to the sparsity of K, it may be possible that the eigenvalues of $A - BK$ cannot be assigned to the desired locations given in $\mathcal{P}$. In fact, full characterization of determining whether pole placement under sparsity constraints is feasible or not does not exist in the literature, although partial analysis exists [8]. Thus, for tractability, we make the following assumption on the feasibility of **Sparse-MGPP** problem.

Assumption 2. *There exists a K satisfying constraints* (6a) *and* (6b)*.*

The **MGPP** and **Sparse MGPP** problems have been studied when the system model (matrices A, B) is known. However, there may be scenarios where (A, B) are not known or cannot be modeled [11], but the system is available for experimentation. In this case, input and state data can be generated by probing the system over time. The aim is to use this data to solve the problems mentioned above.

In this paper, we study the data-driven formulations and solutions of the **MGPP** and **Sparse-MGPP** problems based on the input-state trajectories generated by the open-loop system (1). The proposed methods are direct, that is, no system identification (of matrices (A, B)) is required.

III. Data-driven PP, MGPP and Sparse-MGPP Problems

For the data-driven formulation, we assume the availability of the input-state trajectory data $(u_{[0,T-1]}, x_{[0,T]})$ generated by open-loop system (1). Here, T is the time horizon until which the data is recorded. Using the trajectory data, we construct the following data matrices:

$$U_0 = \begin{bmatrix} u(0) & u(1) & \cdots & u(T-2) & u(T-1) \end{bmatrix},$$
$$X_0 = \begin{bmatrix} x(0) & x(1) & \cdots & x(T-2) & x(T-1) \end{bmatrix},$$
$$X_1 = \begin{bmatrix} x(1) & x(2) & \cdots & x(T-1) & x(T) \end{bmatrix}.$$

Since the data is collected from system (1), the matrices satisfy the relation $X_1 = AX_0 + BU_0$. In the data-driven analysis, it is essential to ensure that the collected data is rich and sufficiently informative such that any trajectory can be expressed as a linear combination of the recorded data [11]. Hence, we make the following assumption on the persistency of excitation.

Assumption 3. *The data matrices U_0 and X_0 satisfy the following rank condition*

$$\textit{Rank}\left(\begin{bmatrix} X_0 \\ U_0 \end{bmatrix}\right) = n + m.$$

A. Pole Placement (PP) Problem

The equation (3) provides conditions for pole placement in terms of system matrices (A, B). Next, we present equivalent conditions in terms of the system data.

Lemma 1. ***(Data-driven pole placement)*** *Let Assumption 3 hold true. Then, there exist (K, X) satisfying the conditions in* (3) *if and only if there exists a $G \in \mathbb{C}^{T\times n}$ satisfying*

$$X_1 G = X_0 G\Lambda, \;\; \textit{Rank}(G) = \textit{Rank}(X_0 G) = n. \tag{7}$$

Further, if such a G exists, then $K = -U_0 G(X_0 G)^{-1}$ and $X = X_0 G$ achieve pole placement.

Proof. From Theorem 2 in [11], we have that all feedback and closed-loop matrices can be parametrized as $K = -U_0 M$ and $A - BK = X_1 M$, respectively, where $M \in \mathbb{R}^{T\times n}$ is a full column rank matrix satisfying $X_0 M = I$. Thus, existence of (K, X) satisfying (3) is equivalent to existence of (M, X) satisfying

$$X_1 MX = X\Lambda, X_0 M = I, \text{Rank}(M) = \text{Rank}(X) = n. \tag{8}$$

In this case, $K = -U_0 M$ archives the pole placement. Next, we reparametrize the conditions in (8) by substituting $G = MX$, which yields $M = GX^{-1}$. Following this reparametrization, the second equality in (8) becomes $X = X_0 G$. Since X is full rank, this necessitates that G be full rank. Substituting $X = X_0 G$ in the first equality in (8), we get that existence of (M, X) satisfying (8) is equivalent to the existence of G satisfying (7). □

Remark 1. ***(Comparison with [13])*** *The pole placement characterization in* (7) *is same as presented in Theorem 4.1 in [13]. However, a couple of comments are in order. First, the parameter M in [13] is real, thereby resulting in a real eigenvector matrix $X = X_0 M$. Thus, their formulation will not work when the desired eigenvalues are complex since it results in complex eigenvectors. In contrast, our parameter G is complex and supports complex eigenvalue assignment. Second, our derivation directly uses the parametrization presented in [11] and a Sylvester equation-based parametrization*[1]*. Note that the former utilises the condition of persistently excited data (data-based) [16] whereas the latter is a model-based equation parameterized to obtain a single free variable G. Therefore, this alternate derivation neatly connects these two fundamental results.*

[1]In this, the parametrization $G = KX$ is used in (3) to get the Sylvester equation $AX - X\Lambda = BG$. Then, G is considered a free variable, and corresponding invertible X is obtained from the above Sylvester equation and feedback matrix is obtained as $K = GX^{-1}$.

Next, we discuss the feasibility of (7).

Lemma 2. ***(Feasibility of pole placement)*** *Let Assumption 3 hold true. Define $\mathcal{N}_i = \textit{Null}\left(\begin{bmatrix} A - \lambda_i I & B \end{bmatrix}\right) \subset \mathbb{C}^{m+n}$. Then,*
(i) there always exists a G satisfying $X_1 G = X_0 G\Lambda$,
(ii) $\textit{Rank}(G) = \textit{Rank}(X_0 G) = n$ if and only if there exist linearly independent vectors $\{y_i\}_{i=1}^n, y_i \in \mathbb{C}^n$ that satisfy $\begin{bmatrix} y_i \\ z_i \end{bmatrix} \in \mathcal{N}_i$ for some $z_i \in \mathbb{C}^m$.

Proof. (i) Let the columns of G be $G = [g_1, g_2, \cdots, g_n]$. Then, $X_1 G = X_0 G\Lambda$ can be decomposed column-wise as n equations:

$$\begin{aligned} X_1 g_i &= X_0 g_i \lambda_i \qquad i = 1, 2, \cdots, n \\ \Rightarrow \begin{bmatrix} A - \lambda_i I & B \end{bmatrix} &\begin{bmatrix} X_0 \\ U_0 \end{bmatrix} g_i = 0 \\ \Rightarrow &\begin{bmatrix} X_0 \\ U_0 \end{bmatrix} g_i \in \mathcal{N}_i, \end{aligned} \tag{9}$$

where we have used the property $X_1 = AX_0 + BU_0$. Since $\text{Range}\left(\begin{bmatrix} X_0 \\ U_0 \end{bmatrix}\right) = \mathbb{R}^{m+n}$, there always exists $g_i \in \mathbb{C}^T$ satisfying (9) for any $\mathcal{N}_i$.
(ii) Suppose there exist linearly independent vectors $\{y_i\}_{i=1}^n, y_i \in \mathbb{C}^n$ that satisfy $\begin{bmatrix} y_i \\ z_i \end{bmatrix} \in \mathcal{N}_i$. Then, set $X_0 g_i = y_i$ and note that such a g_i always exist since $\text{Range}(X_0) = \mathbb{R}^n$. This results in $X_0 G = \begin{bmatrix} y_1, y_2, \cdots, y_n \end{bmatrix}$ to be of full rank n. Finally, since $\text{Rank}(X_0 G) = n$, this implies that $\text{Rank}(G) = n$. The converse statement can be proved similarly. □

Lemma 2 shows that the existence of full rank $X_0 G$ depends on the geometry of the null spaces $\mathcal{N}_i$. A partial analysis of these null spaces is presented in [17], and the authors show that the parameterized variable G has full rank almost always. We defer the complete geometric analysis of the null spaces as a future work.

Next, using the data-based parametrization of the **PP**, we present the data-driven **MGPP** problem.

B. Minimum-Gain Pole Placement (MGPP) Problem

Using Lemma 1, we re-parametrize the **MGPP** problem (4) in terms of parameter G and replace constraint (4a) by (7).

$$\textbf{MGPP}: \min_{G\in\mathbb{C}^{T\times n}} J = \frac{1}{2}\|-U_0 G(X_0 G)^{-1}\|^2 \tag{10}$$

$$\text{s.t.} \quad X_1 G = X_0 G\Lambda. \tag{10a}$$

As mentioned before, we have assumed that the rank conditions on G and $X_0 G$ in (7) are satisfied, and therefore, we have omitted them from the above problem. We also remark that in our simulations, we have verified that these rank conditions are always satisfied.

Problem (10) has a linear constraint. However, the cost function contains an inverse term, which makes the problem

non-convex. Thus, it may exhibit multiple local minima. We aim to solve the constrained optimization Problem (10) using the projection-based gradient descent method. Next, we provide an analytical expression of the gradient of the cost in (10).

Theorem 1. ***(Gradient for MGPP)*** *The gradient of the cost in* (10) *with respect to* G *is given by*

$$\nabla_G J = \frac{\partial J}{\partial G} = -((X_0G)^{-1}K^T(U_0 + KX_0))^T, \quad (11)$$
$$\text{where } K = -U_0G(X_0G)^{-1}.$$

Proof. Let $\mathrm{d}(\cdot)$ denote the differential operator. Using the properties $\|Y\|^2 = \mathrm{Tr}(Y^TY)$ and $\mathrm{dTr}(Y^TY) = 2\mathrm{Tr}(Y^T\mathrm{d}Y)$, we get

$$J = \frac{1}{2}\|K\|^2 = \frac{1}{2}\mathrm{Tr}(K^TK) \Rightarrow \mathrm{d}J = \mathrm{Tr}(K^T\mathrm{d}K). \quad (12)$$

Applying the property $\mathrm{d}Y^{-1} = -Y^{-1}\mathrm{d}YY^{-1}$ on $K = -U_0G(X_0G)^{-1}$, we get

$$\mathrm{d}K = -(U_0\mathrm{d}G - U_0G(X_0G)^{-1}X_0\mathrm{d}G)(X_0G)^{-1}. \quad (13)$$

Substituting (13) in (12), we get

$$\begin{aligned}\mathrm{d}J &= -\mathrm{Tr}(K^T(U_0\mathrm{d}G + KX_0\mathrm{d}G)(X_0G)^{-1}) \\ &= -\mathrm{Tr}((X_0G)^{-1}K^T(U_0 + KX_0)\mathrm{d}G).\end{aligned}$$

Let ∇f be the gradient of $f(N) : \mathbb{C}^n \to \mathbb{R}$, then $\mathrm{d}f = (\nabla_N f)^T\mathrm{d}N$, [18] thus (11) follows. □

To maintain feasibility during the gradient descent, we perform a projection on the constraint space at every iteration. Specifically, if G_k is the iterate at time k, then its projection is obtained by solving the following optimization problem

$$\min_{G\in\mathbb{C}^{T\times n}} J_G = \frac{1}{2}\|G - G_k\|^2 \quad (14)$$
$$\text{s.t.} \quad \text{(10a) holds.}$$

The above problem has quadratic cost and linear constraints, hence it is convex. The next result provides its optimal solution.

Lemma 3. ***(Projection)*** *Define following matrices and Kronecker products:*

$$\bar{X}_1 = \begin{bmatrix} X_1^T & 0 \\ 0 & X_1 \end{bmatrix}, \bar{X}_0 = \begin{bmatrix} X_0^T & 0 \\ 0 & X_0 \end{bmatrix},$$
$$\bar{I} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \ \bar{G}_k = \begin{bmatrix} G_k^* \\ 0 \end{bmatrix},$$

$$C_1 = (I \otimes \bar{X}_1 - \Lambda \otimes \bar{X}_0), \ C_2 = I \otimes \bar{I}.$$

Let $L \in \mathbb{C}^{n\times n}$ *be the matrix of Lagrangian multipliers and define* $Q = \begin{bmatrix} L \\ G \end{bmatrix} = Q^R + jQ^I$. *Further, let* $q = q^R + jq^I = vec(Q)$ *and* $\bar{g}_k = \bar{g}_k^R + j\bar{g}_k^I = vec(\bar{G}_k)$. *Then, the solution to Problem* (14) *is given by*

$$G = G^R + jG^I = \begin{bmatrix} 0 & I \end{bmatrix}\left[Mat(q^R) + j\ Mat(q^I)\right], \quad (15)$$

where

$$\begin{bmatrix} q^R \\ q^I \end{bmatrix} = \bar{C}^\dagger \begin{bmatrix} \bar{g}_k^R \\ \bar{g}_k^I \end{bmatrix}, \ \bar{C} = \begin{bmatrix} C_1^R + C_2^R & -C_1^I \\ C_1^I & C_1^R - C_2^R \end{bmatrix}. \quad (16)$$

Proof. Using the property $\|Y\|^2 = \mathrm{Tr}(Y^HY)$, we get

$$J_G = \frac{1}{2}\mathrm{Tr}((G - G_k)^H(G - G_k)).$$

Let $L \in \mathbb{C}^{n\times n}$ be the matrix of Lagrangian multipliers associated with constraint (10a). Then, the Lagrangian function is given by (see [19]):

$$\begin{aligned}\mathcal{L} =& \mathrm{Tr}((G - G_k)^H(G - G_k)) + \mathbf{1}_n^T(L \circ (X_1G - X_0G\Lambda))\mathbf{1}_n \\ &+ \mathbf{1}_n^T(L \circ (X_1G - X_0G\Lambda))^*\mathbf{1}_n \\ =& \mathrm{Tr}((G - G_k)^H(G - G_k)) + \mathrm{Tr}(L^T(X_1G - X_0G\Lambda)) \\ &+ \mathrm{Tr}(L^H(X_1G - X_0G\Lambda)^*).\end{aligned}$$

Differentiating $\mathcal{L}$ with respect to G and equating to 0, we get

$$\frac{\partial \mathcal{L}}{\partial G} = (G - G_k)^* + X_1^TL - X_0^TL\Lambda = 0. \quad (17)$$

Let $Q = \begin{bmatrix} L \\ G \end{bmatrix}$. Combining (10a) and (17) yields

$$\bar{X}_1Q - \bar{X}_0Q\Lambda + \bar{I}Q^* = \bar{G}_k. \quad (18)$$

Next, we vectorize (18) using properties $\mathrm{vec}(AB) = (I \otimes A)\mathrm{vec}(B)$ and $\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$ to get

$$(I \otimes \bar{X}_1 - \Lambda \otimes \bar{X}_0)q + (I \otimes \bar{I})q^* = \bar{g}_k. \quad (19)$$

Expressing (19) in terms of real and imaginary components results in

$$(C_1^R + jC_1^I)(q^R + jq^I) + C_2^R(q^R - jq^I) = \bar{g}_k^R + j\bar{g}_k^I. \quad (20)$$

Now, separating and equating the real and imaginary parts of (20), we get

$$\bar{C}\begin{bmatrix} q^R \\ q^I \end{bmatrix} = \begin{bmatrix} \bar{g}_k^R \\ \bar{g}_k^I \end{bmatrix}. \quad (21)$$

Thus, (16) follows from (21). Finally, since $q^R = \mathrm{vec}(Q^R)$ and $q^I = \mathrm{vec}(Q^I)$, we get

$$G^R = \begin{bmatrix} 0 & I \end{bmatrix}\mathrm{Mat}(q^R), \ G^I = \begin{bmatrix} 0 & I \end{bmatrix}\mathrm{Mat}(q^I),$$

which results in (15). □

Remark 2. *(Implementation of Projection in Algorithm 1) At each iteration, rank conditions for* G *must hold. If they are violated at any instance, we can randomly perturb* G *slightly such that it satisfies the conditions. Note that we have not encountered this scenario in our simulations presented in Section IV.*

Using the gradient and projection expressions in (11) and (15), we implement the projection-based gradient descent Algorithm 1 given below.

In Algorithm 1, we use steepest descent which takes longer to converge in general. To aid the convergence, we use Armijo's rule or backtracking line search to choose the step size instead of setting a fixed step size.

Algorithm 1 MGPP: Projection-based gradient descent

Require: X_0, X_1, U_0
Output: K
Initialize: G
repeat
$\alpha \leftarrow$ Update step size (Armijo's rule)
$G \leftarrow G - \alpha \nabla_G J$ (using (11))
$G \leftarrow$ Projection of G using (15)
until convergence
$K \leftarrow -U_0 G (X_0 G)^{-1}$

C. Approximately Sparse-MGPP Problem

Similar to the **MGPP** problem, we reparametrize the **Sparse-MGPP** problem (6) in terms of variable G to obtain

$$\min_{G \in \mathbb{C}^{T \times n}} \frac{1}{2} \| -U_0 G (X_0 G)^{-1} \|^2 \tag{22}$$
$$\text{s.t.} \quad \text{(10a) holds}, \tag{22a}$$
$$S^c \circ \left(-U_0 G (X_0 G)^{-1} \right) = 0. \tag{22b}$$

Note that the reparametrized sparsity constraint (22b) is nonlinear in G, and hence, computing the projection on this constraint is not tractable. Therefore, we relax the problem by removing the explicit sparsity constraints and modifying the cost function to add a penalty for violating the sparsity constraints. The penalty is added through a weighing matrix W given by

$$W_{ij} = \begin{cases} 1 & \text{if} \quad S_{ij} = 1, \\ \gg 1 & \text{if} \quad S_{ij} = 0. \end{cases}$$

The weight/penalty W_{ij} is large for a sparse entry, thereby forcing K_{ij} to be small.

With the weighing matrix W, the relaxed Approximately-Sparse-MGPP (**AS-MGPP**) optimization problem becomes

$$\textbf{AS-MGPP}: \min_{G \in \mathbb{R}^{T \times n}} J_W = \frac{1}{2} \| W \circ (-U_0 G (X_0 G)^{-1}) \|^2 \tag{23}$$
$$\text{s.t.} \quad \text{(10a) holds}.$$

Similar to the **MGPP** problem (10), we aim to solve the **AS-MGPP** problem using projection-based gradient descent. Note that the constraint in both problems are identical. Hence, we use (15) to compute the projection. The next results provides the gradient of the cost for **AS-MGPP** problem.

Lemma 4. ***(Gradient for AS-MGPP)*** *Let $g \triangleq vec(G)$ and $k \triangleq vec(K)$. Then, the gradient of the cost in* (23) *with respect to g is given by*

$$\nabla_g J_W = -(k^T \bar{W} [(X_0 G)^{-T} \otimes (U_0 + K X_0)])^T, \tag{24}$$

where $\bar{W} = Diag(vec(W \circ W))$.

Proof. Using the properties (i) $\|Y\|^2 = \text{Tr}(Y^T Y) = \text{vec}(Y)^T \text{vec}(Y)$, and (ii) $\text{vec}(Y \circ Z) = \text{vec}(Y) \circ \text{vec}(Z)$, we get

$$J_W = \frac{1}{2} (\text{vec}(W) \circ k)^T (\text{vec}(W) \circ k) = \frac{1}{2} k^T \bar{W} k.$$

Taking differential, we get

$$\mathrm{d}J_W = k^T \bar{W} \mathrm{d}k \tag{25}$$

Next, we derive an expression for $\mathrm{d}k$. Taking the differential on both sides of $K X_0 G = -U_0 G$ and vectorizing, we get

$$\begin{aligned} &\mathrm{d}K (X_0 G) + K X_0 \mathrm{d}G = -U_0 \mathrm{d}G \\ \implies &\mathrm{d}K = -(U_0 + K X_0) \mathrm{d}G (X_0 G)^{-1} \\ \implies &\mathrm{d}k = -[(X_0 G)^{-T} \otimes (U_0 + K X_0)] \mathrm{d}g. \end{aligned} \tag{26}$$

Substituting $\mathrm{d}k$ in (25) and using $\mathrm{d}f = (\nabla_x f)^T \mathrm{d}x$ results in (24). □

The algorithm to solve the **AS-MGPP** is exactly same as Algorithm 1 except that in the gradient-descent step, we use the gradient $\nabla_g J_W$ instead of $\nabla_G J$.

D. Model-based (Indirect) Method for Pole Placement

The data-driven methods discussed in previous subsections for **PP**, **MGPP** and **AS-MGPP** are direct methods which do not require identification of system model (A, B). An alternate approach would be to first identify (A, B) from the data and then use the existing model-based methods to solve these problems. This is known as the model-based/indirect data-driven approach.

Following Theorem 1 in [11], we can identify the system matrices as

$$\begin{bmatrix} A_d & B_d \end{bmatrix} = X_1 \begin{bmatrix} X_0 \\ U_0 \end{bmatrix}^\dagger, \tag{27}$$

where the subscript d denotes that the matrices are obtained from the data. Once (A_d, B_d) are identified, Algorithm 1 and Algorithm 2 in [8] can be used for solving **MGPP** and **AS-MGPP**, respectively. These algorithms use Sylvester equation-based parametrization. One advantage of this model-based approach is that the resulting optimization problem is unconstrained, and therefore, no projections need to be computed in each iteration. A numerical comparison of model-based and direct approaches is presented in Section IV.

IV. NUMERICAL SIMULATIONS

In this section, we present the numerical validation of our algorithms. We implement the algorithms in MATLAB 2023a. We first present simulations for both **MGPP** and **AS-MGPP** problems, and later compare the performance of model-based and direct approaches for noisy data.

A. Implementation of algorithms for *MGPP* and *AS-MGPP*

We consider the following example from [8] for the **MGPP** problem:

$$A = \begin{bmatrix} -3.7653 & -2.1501 & 0.3120 & -0.2484 \\ 1.6789 & 1.0374 & -0.5306 & 1.3987 \\ -2.1829 & -2.5142 & -1.2275 & 0.2833 \\ -13.6811 & -9.6804 & -0.5242 & 2.9554 \end{bmatrix},$$
$$B = \begin{bmatrix} 1 & 1 & 2 & 5 \\ 1 & 3 & 4 & 2 \end{bmatrix}^T.$$

This is an unstable system with eigenvalues $\lambda(A) = \{-2, -1, 1 \pm 2j\}$. We choose the set of desired closed-loop poles as $\mathcal{P} = \{-2, -1, -0.5 \pm 1j\}$ to shift the two unstable poles to the stable region (left-half plane). We generate input-state data for the open loop system in (1) for $T = 10$ using i.i.d Gaussian input signal to excite the system and assume zero initial conditions. We initialize Algorithm 1 at a feasible G that is obtained as in (9) (refer Lemma 2). At each iteration of the algorithm, pole placement is guaranteed due to projection on the feasible set. As discussed in the previous section, multiple local minima may exist for the **MGPP** problem. To capture the global minima, we run the algorithm multiple times with different initializations. One of the solutions obtained by Algorithm 1 is:

$$K = \begin{bmatrix} -0.0678 & 0.3490 & -0.1571 & 0.4603 \\ 0.2289 & 0.1644 & 0.0358 & -0.0668 \end{bmatrix},$$

with $\|K\| = 0.6694$.

Next, for the **AS-MGPP** problem, we consider the example from [14] with following system matrices and sparsity constraints:

$$A = \begin{bmatrix} 1.1178 & 0.001 & 0.511 & -0.403 \\ -0.051 & 0.661 & -0.011 & 0.061 \\ 0.076 & 0.335 & 0.560 & 0.382 \\ 0 & 0.335 & 0.089 & 0.849 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.004 & -0.087 \\ 0.467 & 0.001 \\ 0.213 & -0.235 \\ 0.213 & -0.016 \end{bmatrix}, \; S = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

This is an unstable system with eigenvalues $\lambda(A) = \{1.22, 1.0049, 0.4206, 0.6025\}$. We choose the set of desired closed-loop poles as $\mathcal{P} = \{-0.3, 0.2, 0.5, .7\}$. To induce sparsity, we set the elements of the weighing matrix $W_{ij} = 100$ for $S_{ij} = 0$. The input-state data is generated in the same manner as explained before. One of the solutions obtained by the algorithm is:

$$K = \begin{bmatrix} 10^{-4} & 2.8133 & 2.71 & 0.423 \\ -2.2621 & 1.305 & 10^{-4} & 1.21 \end{bmatrix},$$

with $\|K\| = 4.8705$. Note that the entries of K corresponding to the zero entries in S are of small magnitude and not exactly zero. For this example, we also show a sample run of algorithms for **MGPP** and **AS-MGPP** in Figure 1. We observe that the $\|K\|$ decreases as the iterations progress and eventually converges to a local minimum. Further, the cost achieved by **AS-MGPP** is higher than **MGPP** since the former is a more constrained problem as compared to the latter.

B. Model-based vs direct methods

In this subsection, we compare the pole placement achieved by model-based and direct methods with noisy data. We generate the noisy data by simulating the following open-loop system:

$$x(t+1) = Ax(t) + Bu(t) + w(t),$$

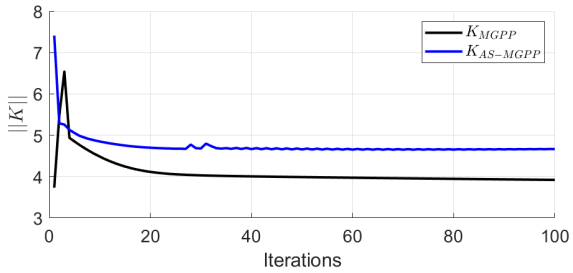


Fig. 1: Sample run of Algorithms for **MGPP** and **AS-MGPP**.

with following random signals: $x(0) \sim \mathcal{N}(0, I), u(t) \sim \mathcal{N}(0.25, 0.5I)$ and $w(t) \sim \mathcal{N}(0, \sigma^2 I)$. We vary n from 2 to 10 and pick m randomly between 1 to n. For each n, we generate a random pair (A, B) ensuring that Assumption 1 holds. The binary sparsity matrix and the number of sparse entries required for **AS-MGPP** is also generated randomly.

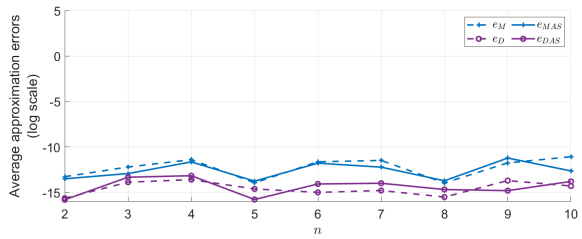
For the model-based method, we identify system matrices (A, B) using (27) and use Algorithms 1 and 2 in [8] for **MGPP** and **AS-MGPP** problems, respectively. The K computed via these methods are denoted as K_M and K_{MAS}, respectively. Similarly, the K computed via data-based methods are denoted as K_D and K_{DAS}, respectively.

To evaluate the effectiveness of pole placement, we compute the error between desired poles and achieved poles using the Hausdroff distance (denoted by $|\cdot|_H$.), which measures the distance between two subsets of a metric space. We calculate the following errors associated with the model-based **MGPP** and **AS-MGPP**: $e_M = |\mathcal{P} - \lambda(A - BK_M)|_H$ and $e_{MAS} = |\mathcal{P} - \lambda(A - BK_{MAS})|_H$. Similarly, for the direct methods, we define $e_D = |\mathcal{P} - \lambda(A - BK_D)|_H$ and $e_{DAS} = |\mathcal{P} - \lambda(A - BK_{DAS})|_H$ associated with **MGPP** and **AS-MGPP**, respectively. Figure 2 shows the pole placement errors for each n averaged over 100 runs. We observe that the errors are considerably small, and direct methods perform better than model-based methods. Further, for larger levels of noise variance, errors observed in model-based methods are larger.

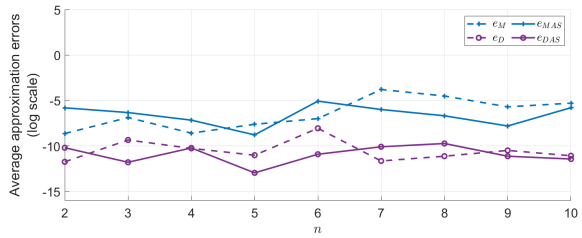
Next, we analyze the accuracy of **MGPP** and **AS-MGPP** algorithms in achieving the state feedback matrix with a lower gain. Table I shows the average norm of K over 100 runs. We observe that the average $\|K\|$ values for the direct case is lower than the model-based case, indicating a better performance by the direct method. The difference between the two increases as the noise level increases. This is because an additional step of system identification from noisy data is required for model-based methods. These experiments provide empirical evidence that direct methods work efficiently with noisy data as well.

V. CONCLUSION

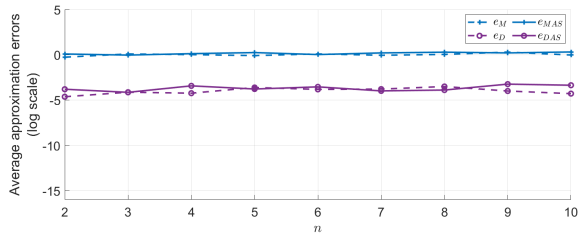
We proposed a projection-based gradient descent method to solve the data-driven minimum-gain pole placement problem in a direct manner. We also studied the sparse version of the problem with sparsity constraints on the feedback matrix and obtained approximately-sparse solutions. Our



(a) Noise variance $\sigma^2 = 0.1$



(b) Noise variance $\sigma^2 = 1$



(c) Noise variance $\sigma^2 = 10$

Fig. 2: Pole placement errors as a function of n for different noise variances.

simulations show that under noisy data, our approach works better than the model-based approach. As future work, we aim to fully characterize the rank conditions on our parametrization based of the geometry of the null spaces of $\begin{bmatrix} A - \lambda_i I & B \end{bmatrix}$. Further, we also aim to solve the exactly-sparse pole placement problem in a data-driven manner without any approximations.

TABLE I: $\|K\|$ for four random pairs of (A, B) at different noise variance values

Non-sparse				
n	Method	Noise variance		
		$\sigma^2 = 0.1$	$\sigma^2 = 1$	$\sigma^2 = 10$
2	Model-based	2.67	4.64	4.19
	Direct	2.37	3.66	3.01
3	Model-based	2.51	2.92	4.32
	Direct	2.41	2.68	3.97
5	Model-based	3.53	7.41	7.92
	Direct	2.85	6.95	7.16
7	Model-based	6.08	6.24	15.07
	Direct	5.98	5.48	12.65
Approximate-sparse				
n	Method	Noise variance		
		$\sigma^2 = 0.1$	$\sigma^2 = 1$	$\sigma^2 = 10$
2	Model-based	4.85	5.85	6.26
	Direct	4.62	5.47	5.43
3	Model-based	3.99	5.44	5.30
	Direct	3.46	4.66	3.85
5	Model-based	4.97	9.89	14.42
	Direct	4.73	9.53	11.54
7	Model-based	8.07	7.28	17.72
	Direct	7.88	6.83	15.50

REFERENCES

[1] W. Wonham, "On pole assignment in multi-input controllable linear systems," *IEEE transactions on automatic control*, vol. 12, no. 6, pp. 660–665, 1967.

[2] R. Schmid, L. Ntogramatzidis, T. Nguyen, and A. Pandey, "A unified method for optimal arbitrary pole placement," *Automatica*, vol. 50, no. 8, pp. 2150–2154, 2014.

[3] S. Bhattacharyya and E. de Souza, "Pole assignment via sylvester's equation," *Systems & Control Letters*, vol. 1, no. 4, pp. 261–263, 1982.

[4] J. Kautsky, N. K. Nichols, and P. Van Dooren, "Robust pole assignment in linear state feedback," *International Journal of control*, vol. 41, no. 5, pp. 1129–1155, 1985.

[5] A. Varga, "Robust pole assignment via sylvester equation based state feedback parametrization," in *CACSD. Conference Proceedings. IEEE International Symposium on Computer-Aided Control System Design (Cat. No. 00TH8537)*, pp. 13–18, IEEE, 2000.

[6] A. Pandey, R. Schmid, and T. Nguyen, "Performance survey of minimum gain exact pole placement methods," in *2015 European Control Conference (ECC)*, pp. 1808–1812, IEEE, 2015.

[7] Y. J. Peretz, "A randomized approximation algorithm for the minimal-norm static-output-feedback problem," *Automatica*, vol. 63, pp. 221–234, 2016.

[8] V. Katewa and F. Pasqualetti, "Minimum-gain pole placement with sparse static feedback," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3445–3459, 2020.

[9] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially invariant systems," *IEEE Transactions on automatic control*, vol. 47, no. 7, pp. 1091–1107, 2002.

[10] V. Krishnan and F. Pasqualetti, "On direct vs indirect data-driven predictive control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 736–741, IEEE, 2021.

[11] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.

[12] T. M. Maupong and P. Rapisarda, "Data-driven control: A behavioral approach," *Systems & Control Letters*, vol. 101, pp. 37–43, 2017.

[13] G. Bianchin, "Data-driven exact pole placement for linear systems," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 685–690, IEEE, 2023.

[14] F. Celi, G. Baggio, and F. Pasqualetti, "Data-driven eigenstructure assignment for sparse feedback design," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 618–623, IEEE, 2023.

[15] S. Mukherjee and R. R. Hossain, "Data-driven pole placement in LMI regions with robustness guarantees," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 4010–4015, IEEE, 2022.

[16] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[17] R. Schmid, A. Pandey, and T. Nguyen, "Robust pole placement with moore's algorithm," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 500–505, 2013.

[18] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," in *IEE Proceedings H (Microwaves, Optics and Antennas)*, vol. 130, pp. 11–16, IET Digital Library, 1983.

[19] A. Hjorungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2740–2746, 2007.