



Iterative learning parametrization for pointwise pole assignment in linear dynamical systems

Jun Zhou¹ · Tiantian Yan¹

Received: 7 January 2021 / Revised: 13 March 2021 / Accepted: 8 April 2021 / Published online: 27 April 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

By means of iterative learning parametrization for state/output feedback gains in linear time invariant systems, pointwise pole assignment (PPA) is re-formulated and addressed in a complex-domain fashion, whereas implementation issues are also examined. Technical features include: (i) no assumptions other than controllability and observability are needed; (ii) the iterative learning parametrization algorithms are numerically tractable and robust against initial values and matrix uncertainties; (iii) the suggested algorithms are significant for achieving PPA-related control strategies, where data-modeling and data-driving techniques are employed. Numerical examples are included to illustrate the main results.

Keywords Pole assignment · Iterative learning · Trace and determinant derivatives · Controllability · Observability

1 Introduction

As is well known, poles and zeros are substantially related to system dynamics or structural properties, and control performances [1]. Almost all structural and spectral features of linear time-invariant systems are related to poles and zeros defined on their state-space realizations and transfer functions [2–6]. Therefore, how to locate or specify their positions and distribution by means of state and/or output feedback gain parametrization is one of the vital techniques for control design and applications, which is simply called pole assignment. It is by no means an exaggeration to say that pole assignment is a key tool for achieving almost all control objectives and performances. Indeed, in the literature, it is not surprising to see that numerous pole assignment problems are formulated and resolved in various ways; as a matter of fact, it is impossible to exhaust the literature even if we focus only on the classical versions of pole assignment.

Roughly speaking, three types of pole assignment are coined and addressed in diversity of terminologies and notations in continuous- and discrete-time systems. In what

follows, we classify these types into pointwise, regional and pinning pole assignments, with or without constraints. By pointwise pole assignment, it means that the closed-loop poles or eigenvalues are placed to some isolated points for obtaining the expected dynamics and modes [7–13]; pole assignment in this sense is typical as well as classical. By regional pole assignment, it implies that the closed-loop poles are placed into some prescribed region so that some dynamics and features can be obtained, while some controller parametrization freedom is available [14–18]; pole assignment in this sense is relatively new and better reflects engineering applications. By pinning pole assignment, it says that some closed-loop poles are pinned to isolated points or one region, while the other ones are located to another distinctive region [19–22]; pole assignment in this sense can be exploited when complicated control plants are involved or subject to uncertainties and constraints. Besides pole assignment, specification in spectra and structures is reported in [21–28], where time-domain as well as frequency-domain concerns are considered.

Motivated by the existing studies, the paper creates an iterative learning state/output feedback gain parametrization approach, which not only solves pole assignment, but also considers the numerical tractability and robustness against initial values and matrix uncertainties. Moreover, the pointwise pole assignment can be formulated in continuous- and discrete-time alike, though only the former cases are talked about.

✉ Tiantian Yan
crystal_yan1993@163.com

Jun Zhou
katsura@hhu.edu.cn

¹ School of Energy and Electrical Engineering, Hohai University, Nanjing 211100, China

The major contribution is a class of iterative learning algorithms for feedback gain parametrization, which possess the technical advantages: (i) no assumptions other than controllability and observability are needed; (ii) the iterative learning parametrization algorithms are numerically tractable and robust against initial values and model uncertainties; (iii) when PPA-related control strategies are based on data-modeling and data-driving, the suggested algorithms can be employed easily.

Outline. Section 2 collects preliminaries to LTI dynamical systems, including the state-space models and their closed-loop expressions under state/output feedback. Pointwise pole assignment problems are concerned in Sect. 3. Technical issues involved in the iterative learning algorithms are summarized in Sect. 4. Numerical examples are sketched in Sect. 5, whereas conclusions are given in Sect. 6.

Notations. \mathcal{R} and \mathcal{C} denote the sets of all real and complex numbers, respectively. I_k denotes the $k \times k$ identity matrix. $(\cdot)^T$ means the transpose of a matrix (\cdot) , while $(\cdot)^*$ means the conjugate transpose of (\cdot) . Also, $(\cdot)^{-T} = ((\cdot)^{-1})^T$.

2 Preliminaries to LTI multivariable systems

Consider the linear time-invariant dynamical continuous-time system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (1)$$

where $x \in \mathcal{R}^n$, $u \in \mathcal{R}^m$, and $y \in \mathcal{R}^l$, respectively, the state, input and output vectors; $A \in \mathcal{R}^{n \times n}$, $B \in \mathcal{R}^{n \times m}$ and $C \in \mathcal{R}^{l \times n}$ are constant matrices.

Now let us introduce to the system (1) the following static state or output feedback, respectively.

$$u(t) = -Kx(t) + v(t), \quad u(t) = -Ly(t) + v(t) \quad (2)$$

where $v \in \mathcal{R}^m$ is a new input, and $K \in \mathcal{R}^{m \times n}$ and $L \in \mathcal{R}^{m \times l}$ are static gains.

Then, the closed-loop state-space equations are

$$\begin{cases} \dot{x}(t) = [A - BK]x(t) + Bv(t) \\ y(t) = Cx(t) \end{cases} \quad (3)$$

and

$$\begin{cases} \dot{x}(t) = [A - BLC]x(t) + Bv(t) \\ y(t) = Cx(t) \end{cases} \quad (4)$$

Correspondingly, we write the characteristic polynomial for (3) and (4), respectively, by

$$\begin{cases} p(s, K) =: \det(sI_n - A + BK) \\ q(s, L) =: \det(sI_n - A + BLC) \end{cases} \quad (5)$$

where $s \in \mathcal{C}$ is the Laplace variable.

Controllability/observability, stabilization and stability of the system (1) are standard and can be found in textbooks about systems and control theory, say [1]. The PBH criteria are important in understanding the discussions, which are referred to Corollaries 12.6.19 and 12.8.4 of [2], respectively.

To facilitate our discussion, let Ω_0 be a simple and close complex contour defined on the complex plane \mathcal{C} (for example, the unit circle for the discrete-time systems; the Nyquist contour for the continuous-time cases). $\{s_0, s_1, \dots, s_\kappa\}$ is a set of isolated points on Ω_0 , namely $s_i \in \Omega_0$, with $i = 0, 1, \dots, \kappa$; $\kappa > 0$ is an integer for the total points number. The interior with Ω_0 being the boundary is denoted by $\text{Int}(\Omega_0)$.

Let us denote the real polynomial

$$\alpha(s) = \alpha_n s^n + \alpha_{n-1} s^{n-1} + \dots + \alpha_1 s + \alpha_0 \quad (6)$$

whose roots belong to $\text{Int}(\Omega_0)$ and are specified according to some expected performances; or $\alpha(s)$ is the desired characteristic polynomial.

3 Pointwise pole assignment via iterative learning algorithm

3.1 Problem 1: pole assignment

The problem in the state feedback is: determine the state feedback gain $K \in \mathcal{R}^{m \times n}$ such that the error function is minimized over $s \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$.

$$J(s, K) = \frac{1}{2} [p(s, K) - \alpha(s)][p(\bar{s}, K) - \alpha(\bar{s})] \quad (7)$$

Or, the problem in the output feedback is: determine the output feedback gain $L \in \mathcal{R}^{m \times l}$ such that the error function is minimized over $s \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$.

$$J(s, L) = \frac{1}{2} [q(s, L) - \alpha(s)][q(\bar{s}, L) - \alpha(\bar{s})] \quad (8)$$

Remark 1 Clearly $J(s, K) > 0$ for any $s \in \Omega_0$ and K . By the error function (7), when the minimum is reached, it holds that $p(s, K) = \alpha(s)$. This says that by installing the state feedback K to the system (1), the closed-loop eigenvalues are assigned to the expected positions. To avoid redundancy, the discussion about the output feedback case is omitted. This is also the case throughout the subsequent arguments.

3.2 Problem 2: pole assignment with gain trace

The gain trace taken into account, the problem in the state feedback is: determine the state feedback gain $K \in \mathcal{R}^{m \times n}$ such that the following error function is minimized over $s \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$.

$$J(s, K) = \frac{1}{2}[p(s, K) - \alpha(s)][p(\bar{s}, K) - \alpha(\bar{s})] + \frac{1}{2}\text{trace}(K^T K) \tag{9}$$

Or, the problem in the output feedback when taking the gain trace into consideration is: determine the output feedback gain $L \in \mathcal{R}^{m \times l}$ such that the following error function is minimized over $s \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$.

$$J(s, L) = \frac{1}{2}[q(s, L) - \alpha(s)][q(\bar{s}, L) - \alpha(\bar{s})] + \frac{1}{2}\text{trace}(L^T L) \tag{10}$$

Remark 2 By (9), when the minimum is reached, it holds that $p(s, K) \rightarrow \alpha(s)$ and $\text{trace}(K^T K) \rightarrow \min$. This says by roots continuity with respect to polynomial coefficients that installing the state feedback K to the system (1), the closed-loop eigenvalues are assigned to some neighborhood around the expected positions, while the state feedback gain is the smallest.

3.3 Problems solving via iterative learning parametrization

In this section, we discuss algorithms for the pole assignment by means of an iterative learning approach. The algorithms present us with numerical approximations of the feedback gains. This is a typical feature of the suggested approach.

The iterative learning algorithms in (11) are alternatively given in terms of $J(s, \mathbf{K})$ and $J(s, \mathbf{L})$ (instead of $J(s, K)$ and $J(s, L)$), which will be defined soon in the following for surmounting mathematical difficulties in calculating determinant and trace derivatives with respect to K and L .

$$\begin{cases} \mathbf{K}^{(k+1)} = \mathbf{K}^{(k)} - \beta \frac{\partial J(s^{(k)}, \mathbf{K}^{(k)})}{\partial \mathbf{K}} \\ \mathbf{L}^{(k+1)} = \mathbf{L}^{(k)} - \gamma \frac{\partial J(s^{(k)}, \mathbf{L}^{(k)})}{\partial \mathbf{L}} \\ k = 0, 1, \dots, \kappa, \quad s^{(k)} \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0 \end{cases} \tag{11}$$

where $\beta > 0$ and $\gamma > 0$ are the learning ratios, and

$$\begin{aligned} \frac{\partial J(s^{(k)}, \mathbf{K}^{(k)})}{\partial \mathbf{K}} &= \left. \frac{\partial J(s, \mathbf{K})}{\partial \mathbf{K}} \right|_{s=s^{(k)}, \mathbf{K}=\mathbf{K}^{(k)}} \\ \frac{\partial J(s^{(k)}, \mathbf{L}^{(k)})}{\partial \mathbf{L}} &= \left. \frac{\partial J(s, \mathbf{L})}{\partial \mathbf{L}} \right|_{s=s^{(k)}, \mathbf{L}=\mathbf{L}^{(k)}} \end{aligned}$$

That is, the partial derivatives are the gradients of $J(s, \mathbf{K})$ and $J(s, \mathbf{L})$ with respect to \mathbf{K} and \mathbf{L} evaluated at $s = s^{(k)}$, and $\mathbf{K}^{(k)}$ and $\mathbf{L}^{(k)}$ (or, the k -th iterative evaluations of \mathbf{K} and \mathbf{L} , respectively).

Since Problems 1 and 2 are substantially the same, we consider to work out a general iterative learning algorithm for solving both of them. More precisely, we will focus only on addressing Problem 2. The iterative learning algorithm to Problem 1 follows as a special case of that to Problem 2.

To this end, let us write $J(s, K)$ in (9) as follows.

$$\begin{aligned} J(s, K) &= \frac{1}{2}[\det([sI_n - A, B] \begin{bmatrix} I_n \\ K \end{bmatrix}) - \alpha(s)] \\ &\quad \cdot [\det([\bar{s}I_n - A, B] \begin{bmatrix} I_n \\ K \end{bmatrix}) - \alpha(\bar{s})] \\ &\quad + \frac{1}{2}\text{trace}(\begin{bmatrix} I_n \\ K \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} I_n \\ K \end{bmatrix}) \\ &= \frac{1}{2}[\det(Q(s)\mathbf{K}) - \alpha(s)][\det(Q(\bar{s})\mathbf{K}) - \alpha(\bar{s})] \\ &\quad + \frac{1}{2}\text{trace}(\mathbf{K}^T E_m \mathbf{K}) =: J(s, \mathbf{K}) \end{aligned} \tag{12}$$

where $Q(s) = [sI_n - A, B] \in \mathcal{C}^{n \times (n+m)}$ and

$$\begin{cases} \mathbf{K} =: \begin{bmatrix} I_n \\ K \end{bmatrix} \in \mathcal{R}^{(n+m) \times n} \\ E_m = \begin{bmatrix} 0 & 0 \\ 0 & I_m \end{bmatrix} \in \mathcal{R}^{(n+m) \times (n+m)} \end{cases}$$

To understand the iterative learning algorithm for fixing \mathbf{K} , we derive the partial derivative of $J(s, \mathbf{K})$ with respect to \mathbf{K} . We observe that

$$\begin{aligned}
 \frac{\partial J(s, \mathbf{K})}{\partial \mathbf{K}} &= \frac{1}{2} \frac{\partial}{\partial \mathbf{K}} [\det(Q(s)\mathbf{K}) - \alpha(s)] \cdot [\det(Q(\bar{s})\mathbf{K}) \\
 &- \alpha(\bar{s})] + \frac{1}{2} [\det(Q(s)\mathbf{K}) - \alpha(s)] \\
 &\cdot \frac{\partial}{\partial \mathbf{K}} [\det(Q(\bar{s})\mathbf{K}) - \alpha(\bar{s})] \\
 &+ \frac{1}{2} \frac{\partial}{\partial \mathbf{K}} [\text{trace}(\mathbf{K}^T E_m \mathbf{K})] \\
 &= \frac{1}{2} \frac{\partial}{\partial \mathbf{K}} [\det(Q(s)\mathbf{K})] \cdot [\det(Q(\bar{s})\mathbf{K}) - \alpha(\bar{s})] \\
 &+ \frac{1}{2} [\det(Q(s)\mathbf{K}) - \alpha(s)] \cdot \frac{\partial}{\partial \mathbf{K}} [\det(Q(\bar{s})\mathbf{K})] \\
 &+ \frac{1}{2} \frac{\partial}{\partial \mathbf{K}} [\text{trace}(\mathbf{K}^T E_m \mathbf{K})] \\
 &= \frac{1}{2} \det(Q(s)\mathbf{K}) Q^T(s)(Q(s)\mathbf{K})^{-T} \cdot [\det(Q(\bar{s})\mathbf{K}) - \alpha(\bar{s})] \\
 &+ \frac{1}{2} [\det(Q(s)\mathbf{K}) - \alpha(s)] \\
 &\cdot \det(Q(\bar{s})\mathbf{K}) Q^T(\bar{s})(Q(\bar{s})\mathbf{K})^{-T} + \frac{1}{2} (E_m + E_m^T) \mathbf{K} \\
 &= \frac{1}{2} \overline{[\det(Q(s)\mathbf{K}) - \alpha(s)] \cdot \det(Q(\bar{s})\mathbf{K}) Q^T(\bar{s})(Q(\bar{s})\mathbf{K})^{-T}} \\
 &+ \frac{1}{2} [\det(Q(s)\mathbf{K}) - \alpha(s)] \\
 &\cdot \det(Q(\bar{s})\mathbf{K}) Q^T(\bar{s})(Q(\bar{s})\mathbf{K})^{-T} + E_m \mathbf{K} \\
 &= \text{Re}([\det(Q(s)\mathbf{K}) - \alpha(s)] \\
 &\cdot \det(Q(\bar{s})\mathbf{K}) Q^T(\bar{s})(Q(\bar{s})\mathbf{K})^{-T}) + E_m \mathbf{K} \tag{13}
 \end{aligned}$$

Similarly, we observe the following algebras.

$$\begin{aligned}
 J(s, L) &= \frac{1}{2} [\det([sI_n - A, B] \begin{bmatrix} I_n & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} I_n \\ C \end{bmatrix}) - \alpha(s)] \\
 &\cdot [\det([\bar{s}I_n - A, B] \begin{bmatrix} I_n & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} I_n \\ C \end{bmatrix}) - \alpha(\bar{s})] \\
 &+ \frac{1}{2} \text{trace}(\begin{bmatrix} I_n & 0 \\ 0 & L \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & I_l \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & L \end{bmatrix}) \\
 &= \frac{1}{2} [\det(Q(s)\mathbf{L}\mathbf{C}) - \alpha(s)] [\det(Q(\bar{s})\mathbf{L}\mathbf{C}) - \alpha(\bar{s})] \\
 &+ \frac{1}{2} \text{trace}(\mathbf{L}^T E_l \mathbf{L}) =: J(s, \mathbf{L}) \tag{14}
 \end{aligned}$$

where $E_l \in \mathcal{R}^{(n+l) \times (n+l)}$ is similar to E_m but with I_m being replaced by I_l ; and

$$\mathbf{L} =: \begin{bmatrix} I_n & 0 \\ 0 & L \end{bmatrix} \in \mathcal{R}^{(n+m) \times (n+L)}, \quad \mathbf{C} = \begin{bmatrix} I_n \\ C \end{bmatrix} \in \mathcal{R}^{(n+l) \times n}$$

Now we turn to derive the derivative of $J(s, \mathbf{L})$ with respect to \mathbf{L} . We observe that

$$\begin{aligned}
 \frac{\partial J(s, \mathbf{L})}{\partial \mathbf{L}} &= \frac{1}{2} \frac{\partial}{\partial \mathbf{L}} [\det(Q(s)\mathbf{L}\mathbf{C}) - \alpha(s)] \cdot [\det(Q(\bar{s})\mathbf{L}\mathbf{C}) \\
 &- \alpha(\bar{s})] + \frac{1}{2} [\det(Q(s)\mathbf{L}\mathbf{C}) - \alpha(s)] \cdot \frac{\partial}{\partial \mathbf{L}} [\det(Q(\bar{s})\mathbf{L}\mathbf{C}) \\
 &- \alpha(\bar{s})] + \frac{1}{2} \frac{\partial}{\partial \mathbf{L}} [\text{trace}(\mathbf{L}^T E_l \mathbf{L})] \\
 &= \frac{1}{2} \det(Q(s)\mathbf{L}\mathbf{C}) Q^T(s)(Q(s)\mathbf{L}\mathbf{C})^{-T} \mathbf{C}^T \\
 &\cdot [\det(Q(\bar{s})\mathbf{L}\mathbf{C}) - \alpha(\bar{s})] + \frac{1}{2} [\det(Q(s)\mathbf{L}\mathbf{C}) - \alpha(s)] \\
 &\cdot \det(Q(\bar{s})\mathbf{L}\mathbf{C}) Q^T(\bar{s})(Q(\bar{s})\mathbf{L}\mathbf{C})^{-T} \mathbf{C}^T + \frac{1}{2} (E_l + E_l^T) \mathbf{L} \\
 &= \text{Re}([\det(Q(s)\mathbf{L}\mathbf{C}) - \alpha(s)] \\
 &\cdot \det(Q(\bar{s})\mathbf{L}\mathbf{C}) Q^T(\bar{s})(Q(\bar{s})\mathbf{L}\mathbf{C})^{-T}) + E_l \mathbf{L} \tag{15}
 \end{aligned}$$

In summary, we claim the following pole assignment procedures according to (11) and (13). The procedures for (11) and (15) can be given similarly.

1. Set $k = 0$, $s^{(0)} = s_0$, and initialize $\mathbf{K}^{(0)} \in \mathcal{R}^{(n+m) \times n}$ with the top $n \times n$ sub-matrix being I_n (respectively., $\mathbf{L}^{(0)} \in \mathcal{R}^{(n+m) \times (n+l)}$ with the (1,1)-sub-matrix being I_n , and the (1,2)- and (2,1)-sub-matrices being zeros);
2. Calculate $\left. \frac{\partial J(s, \mathbf{K})}{\partial \mathbf{K}} \right|_{s=s^{(k)}, \mathbf{K}=\mathbf{K}^{(k)}}$ by (13) (respectively., $\left. \frac{\partial J(s, \mathbf{L})}{\partial \mathbf{L}} \right|_{s=s^{(k)}, \mathbf{L}=\mathbf{L}^{(k)}}$ by (15));
3. If $\det(Q(s^{(k)})\mathbf{K}^{(k)}) = 0$, then $Q(s^{(k)})\mathbf{K}^{(k)}$ is not invertible (respectively, if $\det(Q(s^{(k)})\mathbf{L}^{(k)}\mathbf{C}) = 0$, then $Q(s^{(k)})\mathbf{L}^{(k)}\mathbf{C}$ is not invertible), and drop this $s^{(k)}$ by letting $s_k = s_{k+1}$ over $k = k + 1, \dots, \kappa - 1$ and setting $\kappa = \kappa - 1$, and return to the previous step; otherwise, go forward to the next step;
4. Determine $\mathbf{K}^{(k+1)}$ (resp., $\mathbf{L}^{(k+1)}$) according to (11), and let the top $n \times n$ sub-matrix of $\mathbf{K}^{(k+1)}$ be I_n (resp., let the (1,1)-sub-matrix of $\mathbf{L}^{(k+1)}$ be I_n , and the (1,2)- and (2,1)-sub-matrices of $\mathbf{L}^{(k+1)}$ be zeros);
5. Test if $\|\mathbf{K}^{(k+1)} - \mathbf{K}^{(k)}\| \leq \epsilon$ (resp., $\|\mathbf{L}^{(k+1)} - \mathbf{L}^{(k)}\| \leq \epsilon$) is satisfied, where $\epsilon > 0$ is the tolerance error sufficiently small;
6. If yes, go to the next step; otherwise, set $k = k + 1$ and go back to Step. 2;
7. Let $\mathbf{K} = \mathbf{K}^{(k+1)}$ (resp., $\mathbf{L} = \mathbf{L}^{(k+1)}$) and end.

Remarks about the iterative learning pole assignment algorithm:

- The obtained K (resp., L) is quasi-optimal at most, since the iterative learning algorithms are constructed in terms of \mathbf{K} or \mathbf{L} , in which K and L are their sub-matrices.
- When determining the contour Ω_0 , trial-and-errors are inevitable. In principle, Ω_0 should include all the expected closed-loop eigenvalues in its interior; to improve the iterative learning efficiency, Ω_0 needs to be near to the expected eigenvalues.
- In each iteration, the invertibility of $Q(s)\mathbf{K}$ is assumed. Now we see that the invertibility is satisfied if the pair (A, B) is controllable. Although the iterative learning algorithm does not explicitly entail controllability of the concerned system, it is an underlying assumption for the algorithm to be numerically meaningful. To see this, let us suppose that the pair (A, B) is uncontrollable. This implies that some eigenvalues of $A - BK$ cannot be assigned as expected by choosing the feedback gain K . In other words, $p(s, K) = \alpha(s)$ cannot be true no matter how K is fixed. This in turn says that $J(s, K)$ defined in (9) cannot achieve its minimum.
- The output feedback cannot assign all the closed-loop eigenvalues as expected in general, even if the system (1) is controllable and observable [1]. Bearing this in mind, the suggested algorithm for the output feedback case at most provides us with some partial pole assignment.

Now we claim the following results.

Proposition 1 *Assume that the system (1) is controllable. Consider the iterative learning algorithm in the first equation of (11). If for each $s \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$ with $\kappa > 0$ sufficiently large, $J(s, \mathbf{K}^{(k)}) \rightarrow 0$ as $k \rightarrow \infty$, then all the eigenvalues of $A - BK^{(k)}$ are assigned to small neighborhoods around the roots of $\alpha(s) = 0$.*

Proof of Proposition 1 Under the given assumptions, $J(s, \mathbf{L}^{(k)}) \rightarrow 0$ as $k \rightarrow \infty$ implies that $J(s, K^{(k)}) \rightarrow 0$ as $k \rightarrow \infty$. This can be interpreted as $p(s, K^{(k)}) \rightarrow \alpha(s)$. Then eigenvalue continuity with regard to polynomial coefficients yields the desired assertion. \square

Proposition 2 *Assume that the system (1) is controllable and observable. Consider the iterative learning algorithm in the second equation of (11). If for each $s \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$ with $\kappa > 0$ sufficiently large, $J(s, \mathbf{L}^{(k)}) \rightarrow 0$ as $k \rightarrow \infty$, then it is obvious that $\min\{n, \text{rank } B + \text{rank } C - 1\}$ eigenvalues of $A - BL^{(k)}C$ can be assigned to small neighborhoods around the corresponding roots of $\alpha(s) = 0$.*

4 Numerical issues about the iterative learning algorithms

When implementation of the iterative learning algorithms is concerned, there are numerical issues that need to be explained carefully.

This section is written for collecting such issues and give solutions, if any. This section can be simply skipped if the reader has no interest in the numerical aspects of the iterative learning pole assignment algorithms.

4.1 Convergence and learning ratios

In the above, iteration convergence in the suggested algorithms is not examined, and the learning ratios are constants. Since these problems are somehow beyond the scope of this study, these issues are left for our subsequent studies. It must be added that as far as the authors are concerned, convergence of almost all iterative learning algorithms are still open problems.

4.2 Contours, pole distribution and gain realness

For the obtained state or output feedback gain parametrization to make sense, they must be some real matrices. After carefully reviewing these algorithms, it is not difficult to see that the feedback gains derived do give us with real gain matrices. In what follows, we collect rules about how to avoid complex feedback gains.

- The contour must be symmetric with respect to the real axis, besides being simple and close;
- The specified poles must be real ones or in pairs of conjugate complex numbers.

4.3 Alternative evaluation functions and summation iterative learning

Based on the error evaluation functions defined in (7), (8), (9) and (10), the iterative learning algorithms of (11) are developed. Carefully examining the definition formula, one can easily find that the algorithms are actually implemented in a point-by-point fashion over each $s^{(k)} \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$. Our numerical simulations show that the iterative learning algorithms in this way are sensitive to the distribution pattern and computational ordering of $s^{(k)} \in \{s_0, s_1, \dots, s_\kappa\} \subset \Omega_0$.

To reduce computational sensitivity, we define the error evaluation functions by

$$\begin{aligned} & \sum_{s^{(k)} \in \{s_0, s_1, \dots, s_\kappa\}} J(s^{(k)}, \mathbf{K}) \\ & \sum_{s^{(k)} \in \{s_0, s_1, \dots, s_\kappa\}} J(s^{(k)}, \mathbf{L}) \end{aligned}$$

which are summations of the error evaluation functions as appropriately. Accordingly, the iterative learning algorithms can be given in form of

$$\begin{cases} \mathbf{K}^{(k+1)} = \mathbf{K}^{(k)} - \beta \sum_{\mathbf{v}_s^{(k)}} \frac{\partial J(s^{(k)}, \mathbf{K}^{(k)})}{\partial \mathbf{K}} \\ \mathbf{L}^{(k+1)} = \mathbf{L}^{(k)} - \gamma \sum_{\mathbf{v}_s^{(k)}} \frac{\partial J(s^{(k)}, \mathbf{L}^{(k)})}{\partial \mathbf{L}} \\ k = 0, 1, 2, \dots \end{cases} \quad (16)$$

In other words, the corresponding iterative learning algorithms in (16) will be implemented in each iteration with the error evaluation summation at all the testing points being summed up, and the iterative algorithm goes repeatedly until the error evaluation summation is sufficiently small.

5 Numerical illustrations

5.1 Example system description

Consider the example system of the reference [18]:

$$G(s) = \frac{\begin{bmatrix} -4 & 0 & -2 & 0 & 0 \\ 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & -2 & 0 & -1 \\ 0 & -2 & 0 & -1 & 0 \\ 3 & 0 & -2 & 0 & -1 \end{bmatrix} \left\| \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right.}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \left\| \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right.}$$

Clearly, $n = 5, m = 3$ and $l = 2$. And the eigenvalues of A are: $\lambda_1 = 0.4009, \lambda_{2,3} = -3.7004 \pm 1.1286j$ and $\lambda_{4,5} = -1.5000 \pm 1.9365j$.

To facilitate our statements, let us write the controllability matrix as follows.

$$Q(s) = [sI_5 - A, B]$$

It is easy to see that $\text{rank}\{Q(s)\} = 5$ for any $s \in \mathcal{C}$ so that by the Popov-Belevitch-Hautus criteria [2], the system is completely controllable.

5.2 Numerical simulations with the proposed iterative learning algorithms

In what follows, only the state feedback cases are considered and the summation iterative learning algorithms in (16) are adopted for the sake of brevity. Throughout the numerical simulations, the learning tolerance error is $\epsilon = 0.0001$. The specified closed-loop eigenvalues are assigned at $\lambda_1 = -1, \lambda_2 = -1.5, \lambda_3 = -2$ and $\lambda_{4,5} = -2.5 \pm j\sqrt{3}$. That is, the expected characteristic polynomial is

$$\alpha(s) = (s + 1)(s + 1.5)(s + 2)(s^2 + 4s + 7)$$

The set Ω_0 consists of the 36 equitably and symmetrically chosen isolated points on the circle with radius r and centered at $(-2, j0)$.

Case 1: $r = 3.2, \beta = 0.12$ with gain trace minimization
The obtained state gain is

$$K = \begin{bmatrix} -1.1335 & 0.3762 & -1.2513 & 0.5784 & 0.6308 \\ -0.0096 & -0.2688 & 0.0442 & 0.0140 & -0.0149 \\ 0.5010 & -0.3567 & -0.6497 & -0.0400 & 0.2433 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$-2.5000 \pm 1.7319i, -0.9988 + 0.0000i \\ -1.5032 + 0.0000i, -1.9980 + 0.0000i$$

Case 2: $r = 3.2, \beta = 0.12$ without gain trace minimization
The obtained state gain is

$$K = \begin{bmatrix} -1.1339 & 0.3761 & -1.2513 & 0.5785 & 0.6311 \\ -0.0094 & -0.2698 & 0.0440 & 0.0145 & -0.0148 \\ 0.5021 & -0.3565 & -0.6504 & -0.0403 & 0.2434 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$-2.5000 \pm 1.7321i, -1.0000 + 0.0000i \\ -1.5000 + 0.0000i, -2.0000 + 0.0000i$$

Case 3: $r = 3, \beta = 0.1$ with gain trace minimization
The obtained state gain is

$$K = \begin{bmatrix} -1.1410 & 0.3732 & -1.2469 & 0.5794 & 0.6348 \\ -0.0134 & -0.2830 & 0.0449 & 0.0191 & -0.0162 \\ 0.5081 & -0.3589 & -0.6452 & -0.0421 & 0.2487 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$-2.5000 \pm 1.7319i, -0.9988 + 0.0000i \\ -1.5032 + 0.0000i, -1.9980 + 0.0000i$$

Case 4: $r = 3, \beta = 0.1$ without gain trace minimization
The obtained state gain is

$$K = \begin{bmatrix} -1.1417 & 0.3730 & -1.2468 & 0.5795 & 0.6354 \\ -0.0131 & -0.2847 & 0.0445 & 0.0198 & -0.0161 \\ 0.5100 & -0.3585 & -0.6463 & -0.0426 & 0.2489 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$-2.5000 \pm 1.7321i, -1.0000 + 0.0000i \\ -1.5000 + 0.0000i, -2.0000 + 0.0000i$$

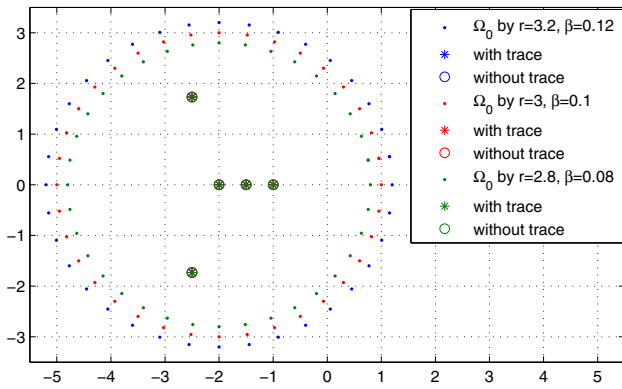


Fig. 1 Pole distributions under iterative learning state feedback pole assignment

Case 5: $r = 2.8, \beta = 0.08$ with gain trace minimization
The obtained state gain is

$$K = \begin{bmatrix} -1.1485 & 0.3716 & -1.2413 & 0.5790 & 0.6404 \\ -0.0163 & -0.3029 & 0.0467 & 0.0224 & -0.0177 \\ 0.5147 & -0.3642 & -0.6402 & -0.0433 & 0.2545 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$\begin{aligned} & -2.5000 \pm 1.7319i, -0.9988 + 0.0000i \\ & -1.5032 + 0.0000i, -1.9980 + 0.0000i \end{aligned}$$

Case 6: $r = 2.8, \beta = 0.08$ without gain trace minimization
The obtained state gain is

$$K = \begin{bmatrix} -1.1497 & 0.3712 & -1.2411 & 0.5791 & 0.6414 \\ -0.0157 & -0.3058 & 0.0461 & 0.0236 & -0.0175 \\ 0.5177 & -0.3636 & -0.6419 & -0.0441 & 0.2548 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$\begin{aligned} & -2.5000 \pm 1.7321i, -1.0000 + 0.0000i \\ & -1.5000 + 0.0000i, -2.0000 + 0.0000i \end{aligned}$$

The set Ω_0 as well as the above numerical pole distributions are plotted in Fig. 1, where the asterisks stand for the closed-loop eigenvalues assigned with gain trace minimization, and the circles represent the ones assigned without gain trace minimization.

Based on Fig. 1, we can observe that

- In each case, the iterative learning algorithm is convergent;
- In general, the gain K obtained with gain trace minimization is slightly smaller than that obtained without gain trace minimization;

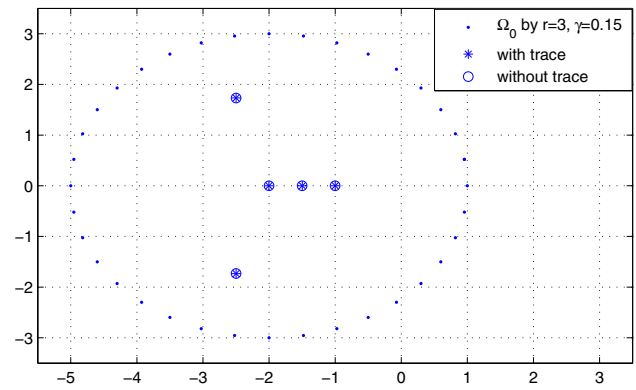


Fig. 2 Pole distributions under iterative learning output feedback pole assignment

- The choice of Ω_0 does have effect on the gain, though its effect on the closed-loop eigenvalues may be negligibly small.

Case 7: $r = 3, \gamma = 0.15$ with gain trace minimization
The obtained output gain is

$$L = \begin{bmatrix} -1.7278 & 1.2278 \\ -0.4253 & -0.1029 \\ 3.8763 & 2.9934 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$\begin{aligned} & -2.4999 \pm 1.7323i, -0.9999 + 0.0000i \\ & -1.4983 + 0.0000i, -2.0020 + 0.0000i \end{aligned}$$

Case 8: $r = 3, \beta = 0.15$ without gain trace minimization
The obtained output gain is

$$L = \begin{bmatrix} -1.7273 & 1.2273 \\ -0.4173 & -0.1023 \\ 3.8852 & 2.9894 \end{bmatrix}$$

and the closed-loop eigenvalues are

$$\begin{aligned} & -2.5000 \pm 1.7321i, -1.0000 + 0.0000i \\ & -1.5000 + 0.0000i, -2.0000 + 0.0000i \end{aligned}$$

The points of the set Ω_0 as well as the above numerical poles distributions are plotted in Fig. 2. Compared to the numerical results of Fig. 1, it is clear to see that the output feedback pole assignment is equally efficient and accurate.

Case 9: $r = 3, \beta = 0.1$ with gain trace minimization, while K and A are randomly set More precisely, the points of the set Ω_0 are the same as those in Fig. 2, and the learning ratio is $\beta = 0.1$. All the entries of K are randomly prescribed initially (namely K_0 is randomly given), while each and all entries of A are perturbed by white noise in form of $A +$

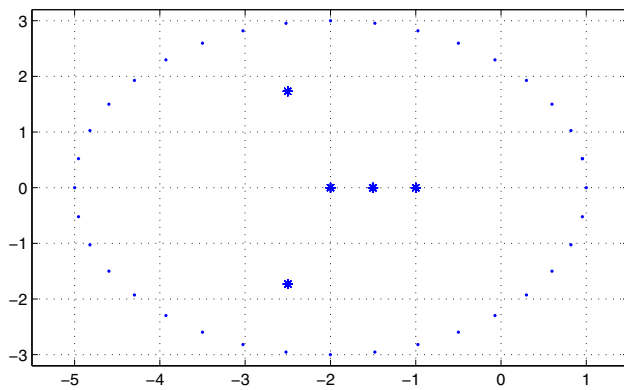


Fig. 3 Pole distributions under iterative learning pole assignment while K and A are randomly perturbed

ΔA with ΔA being randomly given. The learning iteration number is fixed at 4000.

All the numerical poles are plotted in Fig. 3. Compared to the numerical results of Fig. 1, it is clear to see that the state feedback pole assignment is fairly robust with respect to the initial conditions K_0 of the state feedback gain K as well as the perturbations ΔA in the state matrix A . In other words, the suggested pole assignment algorithm is numerically efficient and not sensitive to model uncertainties.

5.3 Numerical results with the conventional method

In what follows, only the state feedback case is considered and the standard pole assignment algorithm in [29] is adopted to fix the feedback gain matrix. The specified closed-loop eigenvalues are assigned at the same positions mentioned as in Sect. 5.2. That is, $\lambda_1 = -1$, $\lambda_2 = -1.5$, $\lambda_3 = -2$ and $\lambda_{4,5} = -2.5 \pm j\sqrt{3}$. The obtained state feedback gain matrix is

$$K = \begin{bmatrix} -1.5419 & 0.1208 & -1.3512 & 1.2792 & -0.1891 \\ -1.0024 & -1.0105 & -1.1687 & 0.9617 & -0.8183 \\ 3.0556 & 0.4281 & -1.9567 & 0.6715 & -0.0406 \end{bmatrix}$$

which has a matrix norm larger than those obtained by the suggested algorithms.

6 Conclusion

In this article, pole assignment in linear dynamical systems is re-formulated and addressed by developing an iterative learning parametrization approach, by exploiting the matrix and trace/determinant derivative concepts. In other words, the suggested design producers are claimed completely from a frequency-domain viewpoint. Numerical implementation of the algorithms are also explicated. The results obtained through numerical examples show how the approach works, and clearly demonstrate high efficiency of the approach.

The selection of learning ratios and convergence in the suggested iterative algorithms remains an interesting and open problem in our future work. This could improve practicability and accuracy of the proposed algorithms hopefully.

Acknowledgements The study is supported by the National Natural Science Foundation of China under Grant No. 61573001.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

- Hespanha JP (2009) Linear System Theory. Princeton University Press, Oxford
- Bernstein DS (2009) Matrix mathematics: theory, facts and formulas, 2nd edn. Princeton, NJ
- Green M, Limebeer DJN (1995) In: Linear Robust Control. Prentice-Hall, Englewood Cliffs, New Jersey, pp 93–96
- Haddad WM, Chellaboina V (2008) Nonlinear Dynamical Systems and Control: a Lyapunov-Based Approach. Princeton University Press, Oxford
- Khalil H (2000) Nonlinear systems, 3rd edn. Pearson Educational International Inc., New Jersey
- Zhou K, Doyle JC (1998) Essentials of Robust control. Prentice Hall, New York
- Konigorski U (2012) Pole placement by parametric output feedback. Syst Control Lett 61(2):292–297
- Mammadov K (2021) Pole placement parametrization for full-state feedback with minimal dimensionality and range. Int J Control 94(2):382–389
- Peretz Y (2017) On parametrization of all the exact pole-assignment state-feedbacks for LTI systems. IEEE Trans Autom Control 62(7):3436–3441
- Schmid R, Pandey A, Nguyen T (2014) Robust pole placement with Moore's algorithm. IEEE Trans Autom Control 59(2):500–505
- Schmid R, Ntogramatzidis L, Nguyen T, Pandey A (2014) A unified method for optimal arbitrary pole placement. Automatica 50(8):2150–2154
- Schmid R, Ntogramatzidis L, Nguyen T (2016) Arbitrary pole placement with the extended Kautsky-Nicholsvan Dooren parametric form. Int J Control 89(7):1359–1366
- Wang AX, Konigorski U (2013) On linear solutions of the output feedback pole assignment problems. IEEE Trans Autom Control 58(9):2354–2359
- Abdelaziz THS (2013) Robust pole placement for second-order linear systems using velocity-plus-acceleration feedback. IET Control Theor Appl 7(14):1843–1856
- Argha A, Su SW, Savkin A, Celler BG (2018) Mixed H_2/H_∞ -based actuator selection for uncertain polytopic systems with regional pole placement. Int J Control 91(1–3):320–336
- dos Santos JFS, Pellanda PC, Simoes AM (2018) Robust pole placement under structural constraints. Syst Control Lett 116:8–14
- Lin ZW, Liu JZ, Zhang WH, Niu YG Regional pole placement of wind turbine generator systems via a Markovian approach. IET Control Theory Appl 10(15):1771–1781

18. Sahoo PR, Goyal JK, Ghosh S, Naskar AK (2019) New results on restricted static output feedback H_∞ controller design with regional pole placement. *IET Control Theor Appl* 13(8):1095–1104
19. Guo ZC, Cai YF, Qian J, Xu SF (2015) A modified Schur method for robust pole assignment in state feedback control. *Automatica* 52:334–339
20. Guo ZC, Qian J, Cai YF, Xu SF (2016) Refined Schur method for robust pole assignment with repeated poles. *IEEE Trans Autom Control* 61(9):2370–2385
21. Shen DM, Wei MS (2015) The pole assignment for the regular triangular decoupling problem. *Automatica* 53:208–215
22. Wei MS, Wang Q, Cheng XH (2010) Some new results for system decoupling and pole assignment problems. *Automatica* 46(5):937–944
23. Babiarz A, Bانشchikova I, Czornik A, Makarov EK, Niezabitowski M, Popova S (2018) Necessary and sufficient conditions for assignability of the Lyapunov spectrum of discrete linear time-varying systems. *IEEE Trans Autom Control* 63(11):3825–3837
24. Cai YF, Qian J, Xu SF (2012) Robust partial pole assignment problem for high order control systems. *Automatica* 48(7):1462–1466
25. Dao MN, Noll D, Apkarian P (2015) Robust eigenstructure clustering by non-smooth optimization. *Int J Control* 88(8):1441–1455
26. Lavaei J, Sojoudi S, Aghdam AG (2010) Pole assignment with improved control performance by means of periodic feedback. *IEEE Trans Autom Control* 55(1):248–252
27. Rosenwasser EN, Lampe BP, Drewelow W, Jeansch T (2019) Causal polynomial pole assignment and stabilization of SISO strictly causal linear discrete processes. *Int J Control* 92(6):1306–1313
28. Tomas-Rodriguez M, Banks SP (2013) An iterative approach to eigenvalue assignment for nonlinear systems. *Int J Control* 86(5):883–892
29. Ogata K (2011) *Modern control engineering*, 5th edn. Publishing House of Electronics Industry, Beijing