

POLE PLACEMENT VIA OUTPUT FEEDBACK: A METHODOLOGY BASED ON PROJECTIONS

Kaiyang Yang^{*†} and Robert Orsi^{*}

^{*}Research School of Information Sciences and Engineering,
The Australian National University, Canberra ACT 0200,
Australia.

[†]National ICT Australia Limited, Locked Bag 8001,
Canberra ACT 2601, Australia.

{Kaiyang.Yang, Robert.Orsi}@anu.edu.au

Abstract: This paper presents an algorithm for solving static output feedback pole placement problems of the following rather general form: given n subsets of the complex plane, find a static output feedback that places in each of these subsets a pole of the closed loop system. The algorithm presented is iterative in nature and is based on alternating projection ideas. Each iteration of the algorithm involves a Schur matrix decomposition, a standard least squares problem and a combinatorial least squares problem. Computational results are presented demonstrating the effectiveness of the algorithm. *Copyright © 2005 IFAC.*

Keywords: static output feedback, pole placement, stabilization, alternating projections.

1. INTRODUCTION

Consider the following generalized static output feedback pole placement problem.

Problem 1. Given $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and closed subsets $\mathcal{C}_1, \dots, \mathcal{C}_n \subset \mathbb{C}$, find $K \in \mathbb{R}^{m \times p}$ such that

$$\lambda_i(A + BKC) \in \mathcal{C}_i \text{ for } i = 1, \dots, n.$$

Here $\lambda_i(A + BKC)$ denotes the i 'th eigenvalue of $A + BKC$.

Problem 1 encompasses many types of pole placement problems. Indeed by varying the choice of \mathcal{C}_i 's, Problem 1 can for example be specialized to the following problems:

- (1) *Classical pole placement:*

$$\mathcal{C}_i = \{c_i\}, c_i \in \mathbb{C}.$$

- (2) *Stabilization type problems for continuous time and discrete time systems:*

$$\mathcal{C}_1 = \dots = \mathcal{C}_n = \{z \in \mathbb{C} \mid \operatorname{Re}(z) \leq -\alpha\}$$

and

$$\mathcal{C}_1 = \dots = \mathcal{C}_n = \{z \in \mathbb{C} \mid |z| \leq \alpha\}$$

respectively.

- (3) *Relaxed classical pole placement:*

$$\mathcal{C}_i = \{z \in \mathbb{C} \mid |z - c_i| \leq r_i\}.$$

- (4) *Hybrid problems:* for example, problems of the type shown in Figure 1. Here $c \in \mathbb{C}$ and the aim is to place a pair of poles at c and \bar{c} , and to place the rest in the region \mathcal{C} :

$$\mathcal{C}_1 = \{c\}, \mathcal{C}_2 = \{\bar{c}\}, \mathcal{C}_3 = \dots = \mathcal{C}_n = \mathcal{C}.$$

This paper presents an algorithm for Problem 1, which is shown to be equivalent to finding a point

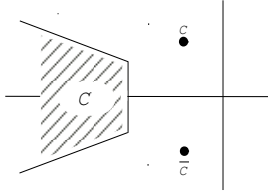


Fig. 1. A hybrid problem.

in the intersection of two particular sets. The algorithm is iterative in nature and is based on an alternating projection like scheme between these two sets. Each iteration of the algorithm involves a Schur matrix decomposition and a standard least squares problem. If the \mathcal{C}_i 's are not all equal, each iteration also requires a combinatorial least squares matching step.

The algorithm is heuristic in nature and convergence to a solution is not guaranteed even if a solution exists. Despite this it will be shown that the algorithm can be quite effective in practice.

Of course, for given (A, B, C) and \mathcal{C}_i 's, Problem 1 may or may not be solvable. Indeed, one would expect that determining whether a particular instance of Problem 1 is solvable is in general difficult. For example, the problem of determining whether the classical pole placement problem is solvable for particular (A, B, C) and desired poles has recently been shown to be NP-hard (Fu, 2004).

In terms of formal requirements placed on the \mathcal{C}_i 's, the main one is that it is possible to calculate projections onto each \mathcal{C}_i . (See section 2 for details on projections.) In particular, the \mathcal{C}_i 's must be closed though they need not be convex or even connected. In addition, for any (A, B, C) and K , each eigenvalue of $A + BKC$ is necessarily real or has a complex conjugate partner and hence, though it is not a formal requirement of the algorithm, it is reasonable to require that each subset \mathcal{C}_i either be symmetric with respect to the real axis or have a complex conjugate partner, that is, a set \mathcal{C}_i' such that $z \in \mathcal{C}_i$ if and only if $\bar{z} \in \mathcal{C}_i'$.

The methods presented here generalize the material presented in (Yang *et al.*, 2004) which dealt solely with the continuous time stabilization problem.

The problem solved by the algorithm is a feasibility problem involving non-symmetric matrices. The idea of solving such problems via alternating projections that utilize Schur's decomposition originates in (Orsi and Yang, n.d.).

There is a great deal of literature dealing with classical pole placement and stabilization. An overview of theoretical results, existing algorithms

and historical develops can be found in (Byrnes, 1989), (Rosenthal and Willems, 1998), (Syrmos *et al.*, 1997), (de Oliveira and Geromel, 1997), (Rosenthal and Sottile, 1998) and (Eremenko and Gabrielov, 2002). As far as the authors are aware, pole placement in the generality presented in Problem 1 has not previously been considered. Note however that, using linear matrix inequality techniques, pole placement in a single convex region has been investigated, see (Chilali and Gahinet, 1996). Lastly, regarding algorithms for classical pole placement, the survey paper (Rosenthal and Willems, 1998) states that existing sufficiency conditions are mainly theoretical in nature and that there are no good numerical algorithms available in many cases when a problem is known to be solvable.

The structure of the paper is as follows. Section 2 introduces projections and how alternating projections can be used to find a point in the intersection of a finite number of closed (convex) sets. Section 3 presents the solution methodology. Section 4 contains computational results of applying the algorithm to various instances of Problem 1. Results are presented for the classical pole placement problem, the discrete time stabilization problem, and a hybrid problem. The paper ends with some concluding remarks.

2. PROJECTIONS

This section introduces projections and the method of alternating projections.

Let x be an element in a Hilbert space H and let \mathcal{D} be a closed (possibly non-convex) subset of H . Any $d_0 \in \mathcal{D}$ such that $\|x - d_0\| \leq \|x - d\|$ for all $d \in \mathcal{D}$ will be called a *projection* of x onto \mathcal{D} . In the cases of interest here, namely that H is a finite dimensional Hilbert space, there is always at least one such point for each x . If \mathcal{D} is convex as well as closed then each x has exactly one such minimum distance point. Any function $P_{\mathcal{D}} : H \rightarrow H$ will be called a *projection operator* (for \mathcal{D}) if for each $x \in H$,

$$P_{\mathcal{D}}(x) \in \mathcal{D} \quad \text{and} \\ \|x - P_{\mathcal{D}}(x)\| \leq \|x - d\| \quad \text{for all } d \in \mathcal{D}.$$

A point in the intersection of a finite number of closed convex sets can be found via alternating projections (Brègman, 1965).

Theorem 2. Let $\mathcal{D}_1, \dots, \mathcal{D}_N$ be closed convex sets in a real finite dimensional Hilbert space H . If $\bigcap_{i=1}^N \mathcal{D}_i$ is nonempty, then starting from an arbitrary initial value, the following sequence

$$x_{i+1} = P_{\mathcal{D}_{\phi(i)}}(x_i), \quad \text{where } \phi(i) = (i \bmod N) + 1, \\ \text{converges to an element in } \bigcap_{i=1}^N \mathcal{D}_i.$$

When one or more \mathcal{D}_i 's are non-convex, Theorem 2 no longer applies and starting the algorithm of Theorem 2 from certain initial values may result in a sequence of points that does not converge to a solution of the problem. However, (Combettes and Trussell, 1990) suggests that alternating projections for problems with one or more non-convex sets converge locally; they will converge if the initial starting point is sufficiently close to a feasible point.

3. METHODOLOGY

This section presents the solution methodology.

From now on $\mathbb{C}^{n \times n}$ will be regarded as a Hilbert space with inner product

$$\langle Y, Z \rangle = \text{tr}(YZ^*) = \sum_{i,j} y_{ij} \bar{z}_{ij}.$$

The associated norm is the Frobenius norm $\|Z\| = \langle Z, Z \rangle^{\frac{1}{2}}$.

Throughout this section it is assumed (A, B, C) and $\mathcal{C}_1, \dots, \mathcal{C}_n$ are given and fixed.

Let \mathcal{L} denote the set of all possible closed-loop system matrices,

$$\mathcal{L} = \{Z \in \mathbb{R}^{n \times n} \mid Z = A + BKC \text{ for some } K \in \mathbb{R}^{m \times p}\},$$

and let \mathcal{M} denote the set of complex matrices with eigenvalues in the specified regions $\mathcal{C}_1, \dots, \mathcal{C}_n$,

$$\mathcal{M} = \{Z \in \mathbb{C}^{n \times n} \mid \lambda_i(Z) \in \mathcal{C}_i, i = 1, \dots, n\}.$$

Problem 1 can now be stated as:

$$\text{Find } X \in \mathcal{L} \cap \mathcal{M}.$$

A solution strategy to solve Problem 1 would be to employ an alternating projection scheme, alternatively projecting between \mathcal{L} and \mathcal{M} . Two difficulties occur in trying to do this. The first is that, while \mathcal{L} is convex, \mathcal{M} is in general a rather complicated set that is non-convex. Hence, alternating projections between \mathcal{L} and \mathcal{M} are not guaranteed to converge. The second, more important issue is that it is not clear how to project onto \mathcal{M} . That is, given a point Z , it is not clear how to find a point in \mathcal{M} of minimal distance to Z .

As will be shown in the experiments, an alternating projection like scheme can still be quite successful if instead of using a true projection map for \mathcal{M} , a suitable substitute is used.

Before proceeding, recall Schur's result (Horn and Johnson, 1985).

Theorem 3. Given $Z \in \mathbb{C}^{n \times n}$ with eigenvalues $\lambda_1, \dots, \lambda_n$ in any prescribed order, there is a

unitary matrix $V \in \mathbb{C}^{n \times n}$ and an upper triangular matrix $T \in \mathbb{C}^{n \times n}$ such that

$$Z = VTV^*,$$

and $T_{kk} = \lambda_k, k = 1, \dots, n$.

The following map is proposed as a substitute for a projection map onto \mathcal{M} . Though it is not a true projection map, the notation $P_{\mathcal{M}}$ will still be used. The choice for $P_{\mathcal{M}}$ is motivated by other substitute maps appearing in (Orsi and Yang, n.d.) and (Yang *et al.*, 2004), which were themselves motivated by true projection maps for certain 'symmetric' problems. In the following, $P_{\mathcal{C}_i}(z)$ will denote the projection of $z \in \mathbb{C}$ onto \mathcal{C}_i .

Definition 4. Suppose $V \in \mathbb{C}^{n \times n}$ is unitary and $T \in \mathbb{C}^{n \times n}$ is upper triangular. Let σ be a permutation of $\{1, \dots, n\}$ such that amongst all possible permutations, it minimizes

$$\sum_{k=1}^n |T_{kk} - P_{\mathcal{C}_{\sigma(k)}}(T_{kk})|^2. \quad (1)$$

Define

$$P_{\mathcal{M}}(V, T) = V\hat{T}V^*$$

where \hat{T} is upper triangular and given by

$$\hat{T}_{kl} = \begin{cases} P_{\mathcal{C}_{\sigma(k)}}(T_{kk}), & \text{if } k = l, \\ T_{kl}, & \text{otherwise.} \end{cases}$$

Note that $P_{\mathcal{M}}$ maps into the set \mathcal{M} . Note also that finding σ involves solving a combinatorial least squares problem. (This will be discussed further later in the section.)

In order to apply $P_{\mathcal{M}}$ to $Z \in \mathbb{C}^{n \times n}$, a Schur decomposition of Z must first be found. A given Z may have a non-unique Schur decomposition and $Z = V_1 T_1 V_1^* = V_2 T_2 V_2^*$ does *not* necessarily imply $P_{\mathcal{M}}(V_1, T_1) = P_{\mathcal{M}}(V_2, T_2)$. Hence, $P_{\mathcal{M}}$ may give different points for different Schur decompositions of the same matrix. This is not so important as different Schur decompositions lead to points in \mathcal{M} of equal distance from the original matrix, as is now shown.

Theorem 5. Suppose $Z = V_1 T_1 V_1^* = V_2 T_2 V_2^*$ where $V_1, V_2 \in \mathbb{C}^{n \times n}$ are unitary and $T_1, T_2 \in \mathbb{C}^{n \times n}$ are upper triangular. Then

$$\|P_{\mathcal{M}}(V_1, T_1) - Z\| = \|P_{\mathcal{M}}(V_2, T_2) - Z\|.$$

Proof. Suppose $Z = VTV^*$ where V is unitary and T is upper triangular. If \hat{T} is the matrix given in Definition 4, then by the unitary invariance of the Frobenius norm,

$$\|P_{\mathcal{M}}(V, T) - Z\| = \|\hat{T} - T\|.$$

As $\|\hat{T}-T\|^2$ equals the quantity in (1), $\|P_{\mathcal{M}}(V,T)-Z\|$ depends only on T_{11}, \dots, T_{nn} (and the sets $\mathcal{C}_1, \dots, \mathcal{C}_n$). The T_{kk} 's are the eigenvalues of Z and hence, aside from ordering, are not decomposition dependent. The result now follows by noting that (1) does not depend on the ordering of the T_{kk} 's. \square

$P_{\mathcal{M}}(V,T)$ keeps V fixed and modifies T . Theorem 6 below shows that of all the points in \mathcal{M} of the form VTV^* , $\tilde{T} \in \mathcal{M}$ upper triangular, i.e., of all the points in \mathcal{M} that have a Schur decomposition with the same V matrix, $P_{\mathcal{M}}(V,T)$ is closest (or at least equal closest) to the original point $Z = VTV^*$.

Theorem 6. Suppose $Z = VTV^* \in \mathbb{C}^{n \times n}$ with V unitary and T upper triangular. Then $P_{\mathcal{M}}(V,T)$ satisfies

$$\|P_{\mathcal{M}}(V,T) - Z\| \leq \|V\tilde{T}V^* - Z\|$$

for all upper triangular $\tilde{T} \in \mathcal{M}$.

Proof. Let \tilde{T} be an upper triangular matrix in \mathcal{M} . The unitary invariance of the Frobenius norm implies the result will be established if it can be shown

$$\|\hat{T} - T\| \leq \|\tilde{T} - T\|,$$

where \hat{T} is the matrix given in Definition 4. As both \tilde{T} and T are upper triangular and $\tilde{T} \in \mathcal{M}$, it follows that

$$\|\tilde{T} - T\|^2 = \sum_{k=1}^n |\tilde{T}_{kk} - T_{kk}|^2 + \sum_{k < l} |\tilde{T}_{kl} - T_{kl}|^2 \quad (2)$$

and that $\tilde{T}_{kk} \in \mathcal{C}_{\tilde{\sigma}(k)}$, $k = 1, \dots, n$, for some permutation $\tilde{\sigma}$.

The result now follows by noting that $\|\hat{T} - T\|^2$ equals the quantity in (1) and that this value must be less than or equal to the first summation on the right hand side of the equality in (2). \square

The projection of $X \in \mathbb{C}^{n \times n}$ onto \mathcal{L} involves solving a standard least squares problem.

Lemma 7. The projection of $X \in \mathbb{C}^{n \times n}$ onto \mathcal{L} is given by $P_{\mathcal{L}}(X) = A+BKC$ where K is a solution of the least squares problem

$$\arg \min_{K \in \mathbb{R}^{m \times p}} \|(C^T \otimes B) \text{vec}(K) - \text{vec}(\text{Re}(X) - A)\|_2.$$

Here $\|\cdot\|_2$ denotes the standard vector 2 norm.

Proof. See (Yang *et al.*, 2004). \square

Here is the algorithm for Problem 1.

Algorithm:

Problem Data. $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $\mathcal{C}_1, \dots, \mathcal{C}_n \subset \mathbb{C}$.

Initialization. Choose a randomly generated matrix $Y \in \mathbb{R}^{n \times n}$. For example, draw each entry of Y from a normal distribution of zero mean and variance 1.

repeat

1. $X := P_{\mathcal{L}}(Y)$.
2. Calculate a Schur decomposition of X : $X = VTV^*$.
3. $Y := P_{\mathcal{M}}(V,T)$.

until $\|X - Y\| < \epsilon$.

Note that as $Y = P_{\mathcal{M}}(V,T) = V\hat{T}V^*$, see Definition 4, $\|X - Y\| = \|VTV^* - V\hat{T}V^*\| = \|T - \hat{T}\| = (\sum |T_{kk} - \hat{T}_{kk}|^2)^{\frac{1}{2}}$. As the T_{kk} 's are the eigenvalues of $X \in \mathcal{L}$ and the \hat{T}_{kk} 's are the eigenvalues of $Y \in \mathcal{M}$, the algorithm stops when X (which equals $A + BKC$ for some K) has eigenvalues sufficiently close a matrix that satisfies the pole placement constraints. In particular, each eigenvalue of such an X cannot violate the pole placement constraints by more than ϵ .

As mentioned previously, $P_{\mathcal{M}}$ involves finding a permutation σ that minimizes (1). The first step in solving this combinatorial least squares problem is calculating the squared distance of each T_{kk} to each subset \mathcal{C}_i and placing these values in a $n \times n$ cost matrix D :

$$D_{kl} := |T_{kk} - P_{\mathcal{C}_l}(T_{kk})|^2. \quad (3)$$

The problem is now equivalent to finding a permutation σ such that $\sum D_{k\sigma(k)}$ is minimal.

The problem of finding a minimizing σ given a cost matrix D is a *linear assignment problem* which can be solved in $O(n^3)$ time using the so called *Hungarian method*, see (Martello and Toth, 1987) for details.

Note that if the \mathcal{C}_i 's are not all distinct, for example this occurs for stabilization problems and what here have been called hybrid problems, the complexity of the matching problem is reduced. In fact for stabilization problems, all the \mathcal{C}_i 's are the same and no matching step is required. For the hybrid problem shown in Figure 1, it is only necessary to check $n(n-1)$ possibilities corresponding to which two T_{kk} 's are matched to c and \bar{c} . Hence for this hybrid problem, the direct approach is faster than using the Hungarian method.

Also note that, given a cost matrix D , an alternative to the Hungarian method is the following faster suboptimal matching strategy. Find the (or a) smallest entry in D , say $D_{\bar{k}\bar{l}}$. Match $T_{\bar{k}\bar{k}}$ with $\mathcal{C}_{\bar{l}}$ and cross out row \bar{k} and column \bar{l} of D . Now only consider the uncrossed out entries in D and repeat, until all n matches have been made. This method does not always find the optimal matching though it can often be quite an effective substitute for the Hungarian method. It will be

termed *suboptimal matching*. Surprisingly, as will be shown in the next section, by using suboptimal matching in the algorithm it was possible to solve a particular problem which was not solvable using the Hungarian method.

4. COMPUTATIONAL RESULTS

This section contains computational results of applying the algorithm to various instances of Problem 1. Results are presented for the classical pole placement problem, the discrete time stabilization problem, and a hybrid problem. (Results for continuous time stabilization appear in (Yang *et al.*, 2004).)

The algorithms for each problem were implemented in Matlab 6.5 and all results were obtained using a 3.06 GHz Pentium 4 machine.

Throughout this section a randomly generated matrix will be a matrix whose entries are drawn from a normal distribution of zero mean and variance 1.

4.1 Classical pole placement: random problems

This subsection contains results for some randomly generated classical pole placement problems. A 1000 problems with $n = 6$, $m = 4$ and $p = 3$ were created. Each problem was created as follows. A, B, C and \tilde{K} were generated randomly. A scalar multiple of the identity was added to A to ensure the stability degree of $A + B\tilde{K}C$ was equal to $\alpha = 0.1$. The desired poles were taken to be the poles of $A + B\tilde{K}C$. Matching was done using the Hungarian method.

An attempt was made to solve each problem using up to 10 different initial conditions and a maximum of a 1000 iterations per initial condition. With the termination parameter ϵ set to $\epsilon = 10^{-3}$, the success rate for 1 initial condition was 50% and the overall success rate was 91%. For the problems that were solved, the average number of iterations taken was 1.8×10^3 and the average time taken was 1.5 CPU seconds.

Note: in (3), if $\mathcal{C}_l = \{c_l\}$ then $P_{\mathcal{C}_l}(T_{kk}) = c_l$.

4.2 Classical pole placement: a particular problem from the literature

The following problem is taken from (Sridhar and Lindorff, 1973) and is of interest as the set of desired poles overlaps with the set of open loop poles. Consider the following system:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}^T.$$

In this problem the aim is to place the closed-loop poles at $\{-1, -2, -3, -5\}$. While initial attempts to solve this problem failed, the problem was solved by using suboptimal matching and replacing Step 3 of the algorithm with ‘ $Y := (1 - \gamma)P_{\mathcal{M}}(V, T) + \gamma X$ ’, $\gamma \in (-1, 1)$ constant. (If $P_{\mathcal{M}}$ were a true projection map, Y would now be what is termed a *relaxed projection* of X onto \mathcal{M} .) Solutions were successfully found by taking γ close to 1. While $\gamma = 0.9, 0.8$ and 0.7 can all be used to successfully find a solution, likelihood of success increases the closer γ is to 1, while speed of convergence decreases. With $\gamma = 0.7$ and $\epsilon = 10^{-3}$, solutions can typically be found in about 1.2×10^4 iterations and about 7.5 CPU seconds. (With ϵ reduced greatly to $\epsilon = 10^{-14}$ a solution was found in about 10^6 iterations and took 275 CPU seconds.)

Note: when employing relaxed projections, the loop termination criterion should be replaced by ‘until $\|X - P_{\mathcal{M}}(V, T)\| < \epsilon$ ’.

4.3 Discrete-time stabilization: random problems

This subsection contains results for some randomly generated discrete time stabilization problems. For each problem, the aim is to place all poles in the set $\mathcal{C} = \{z \mid |z| \leq \alpha\}$, $\alpha = 0.9$. A 1000 (A, B, C) triples with $n = 6$, $m = 4$ and $p = 3$ were randomly generated. Triples with A stable were discarded and replaced.

As in Section 4.1, an attempt was made to solve each problem using up to 10 different initial conditions and a maximum of a 1000 iterations per initial condition. With $\epsilon = 10^{-3}$, the success rate for 1 initial condition was 61% and the overall success rate was 80%. For the problems that were solved, the average number of iterations taken was 3.3×10^3 and the average time taken was 0.37 CPU seconds.

Note: in (3), $P_{\mathcal{C}_l}(T_{kk})$ equals $\frac{\alpha T_{kk}}{|T_{kk}|}$ if $|T_{kk}| \geq \alpha$ and T_{kk} otherwise.

4.4 A hybrid problem

In this subsection a hybrid problem of the type shown in Figure 1 is considered. The problem parameters are $c = -0.5 + i3$, $\mathcal{C} = \{z \in \mathbb{C} \mid \text{Re}(z) \leq -2 \text{ and } |\text{Im}(z)| \leq |\text{Re}(z)|\}$, $n = 13$, $m = 3$ and $p = 5$. To ensure solvability, A was set to $A = \tilde{V}\tilde{T}\tilde{V}^T - B\tilde{K}C$, where B, C and \tilde{K} were randomly generated, \tilde{V} was a randomly chosen orthogonal

matrix, and \tilde{T} was a real block upper triangular matrix whose spectrum satisfied the constraints. (\tilde{T} was assigned the spectrum $\{-0.5 \pm i3, -2, -2 \pm i, -2.3, -2.5, -3 \pm i3, -3.5 \pm i3.1, -4 \pm i4\}$ and was created by choosing appropriate 1×1 and 2×2 blocks for its diagonal. The remaining upper triangular entries of \tilde{T} were chosen randomly.)

With $\epsilon = 10^{-3}$, 64% of initial conditions tested resulted in a solution of this problem within 5000 iterations. For the initial conditions that lead to convergence, the average number of iterations taken was 8.6×10^2 and average time taken was 3.1 CPU seconds. The closed loop poles of a particular solution are shown in Figure 2.

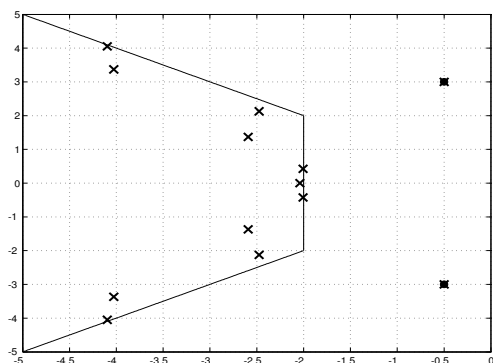


Fig. 2. A hybrid problem: the closed loop pole locations for a particular solution.

Note: the least squares matching steps were done directly rather than using the Hungarian method or suboptimal matching. Due to space limitations, details of how to project onto \mathcal{C} are omitted.

5. CONCLUSION

In this paper a new methodology for solving a broad class of output feedback pole placement problems has been presented. A particular strength of the method is the ability to solve less standard pole placement problems. Presented numerical results demonstrate that the method can be quite effective despite the fact that convergence to a solution is not guaranteed.

6. ACKNOWLEDGEMENT

The second author acknowledges the support of the Australian Research Council through grant DP0450539.

National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

REFERENCES

- Brègman, L. M. (1965). The method of successive projection for finding a common point of convex sets. *Soviet Mathematics* **6**(3), 688–692.
- Byrnes, C. I. (1989). Pole assignment by output feedback. In: *Three decades of mathematical system theory* (H. Nijmeijer and J. M. Schumacher, Eds.). pp. 31–78. Lecture notes in control and inform. sci. 135. Springer Verlag.
- Chilali, M. and P. Gahinet (1996). H_∞ design with pole placement constraints: An LMI approach. *IEEE Trans. Automatic Control* **41**(3), 358–367.
- Combettes, P. L. and H. J. Trussell (1990). Method of successive projections for finding a common point of sets in metric spaces. *J. Optimisation Theory Application* **67**, 487–507.
- de Oliveira, M. C. and J. C. Geromel (1997). Numerical comparison of output feedback design methods. In: *Proc. American Control Conference*. New Mexico. pp. 72–76.
- Eremenko, A. and A. Gabrielov (2002). Pole placement by static output feedback for generic linear systems. *SIAM J. Control Optimization* **41**(1), 303–312.
- Fu, M. (2004). Pole placement via static output feedback is NP-hard. *IEEE Trans. Automatic Control* **49**(5), 855–857.
- Horn, R. A. and C. A. Johnson (1985). *Matrix Analysis*. Cambridge University Press.
- Martello, S. and P. Toth (1987). Linear assignment problems. *Annals of Discrete Mathematics* **31**, 259–282.
- Orsi, R. and K. Yang (n.d.). Numerical methods for solving inverse eigenvalue problems for nonnegative matrices. In preparation.
- Rosenthal, J. and F. Sottile (1998). Some remarks on real and complex output feedback. *Systems and Control Letters* **33**, 73–80.
- Rosenthal, J. and J. C. Willems (1998). Open problems in the area of pole placement. In: *Open problems in mathematical systems and control theory* (V. D. Blondel, E. D. Sontag, M. Vidyasager and J. C. Willems, Eds.). Chap. 37, pp. 181–191. Springer Verlag.
- Sridhar, B. and D. P. Lindorff (1973). Pole placement with constraint gain output feedback. *Int. J. Control* **18**(5), 993–1003.
- Syrmos, V. L., C. T. Abdallah, P. Dorato and K. Grigoriadis (1997). Static output feedback — a survey. *Automatica* **33**(2), 125–137.
- Yang, K., R. Orsi and J. B. Moore (2004). A projective algorithm for static output feedback stabilization. To appear in *Proc. 2nd IFAC Symp. on System, Structure and Control*, Oaxaca, Mexico.