Appears in *Proc. 2014 American Control Conference*, pp. 4298-4305, Portland, OR, June 4-June 6, 2014.

An Optimization Approach to the Pole-Placement Design of Robust Linear Multivariable Control Systems

Richard J. Vaccaro

***Abstract*— Traditional pole-placement methods for calculating state-feedback gains for multivariable regulators or tracking systems do not come with stability robustness guarantees. Even if good robustness happens to be obtained, pole-placement calculation of observer gains for observer-based control systems often results in poor stability margins. In this paper, a parameterization of all feedback gain matrices corresponding to a given set of specified closed-loop pole locations is derived. This parameterization is easily modified for observer gain matrices corresponding to a set of desired observer poles. The feedback or observer gain matrices are calculated by finding the parameters that maximize the H-infinity unstructured stability robustness norm for the given control system. Examples are given showing that the proposed approach yields state feedback regulators with good robustness (better than LQR) and is particularly effective for designing robust observer-based control systems.**

I. POLE PLACEMENT VIA OPTIMIZATION

Consider the calculation of feedback gain matrices via pole placement for a state-feedback control system. If the plant has more than one input then there are an infinite number of feedback gain matrices all of which yield the same closed-loop pole locations. In this case, any pole-placement algorithm makes a choice of some particular gain matrix out of this infinite set. In this paper, the choice is made by optimizing a cost function, which is the input-multiplicative stability robustness norm of the control system.

A similar non uniqueness of gain matrices occurs for full-order observers for a plant having more than one output measurement. In this case there is an infinite set of observer gain matrices all of which yield the same observer pole locations. It is shown in this paper that the algorithm for calculating feedback gain matrices by optimizing robustness may be easily modified for calculating observer gains. The cost function for this optimization is the stability robustness norm of the complete observer-based control system. In what follows, we consider optimization of either feedback gains or observer gains.

If the parameters in the optimization are the entries in the gain matrix itself, the optimization algorithm must satisfy the constraint that any candidate gain matrix yield the desired pole locations. An alternate approach is to derive a parameterization of the set of all gain matrices such that any value of the parameters gives a gain matrix that yields the specified pole locations. These parameters may then be used in an unconstrained optimization of the robustness cost function. The following subsections set up the pole-placement constraint and derive the parameterizations for real- and complex-valued poles.

Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA, vaccaro@ele.uri.edu

A. Pole Placement Constraint

Consider first the calculation of feedback gain matrices. The pole placement constraint may be stated as follows. Given $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ and a set of desired closed-loop pole locations $\Lambda = \{\lambda_1, \cdots, \lambda_n\}$, calculate $\mathbf{K}$ so that the eigenvalues of $\mathbf{A}-\mathbf{B}\mathbf{K}$ are in Λ. The state vector $\mathbf{x}$ contains n state variables and the input vector $\mathbf{u}$ contains p input signals. For ease of exposition we assume that $(\mathbf{A}, \mathbf{B})$ is controllable and that the desired closed-loop pole locations are distinct. Both of these assumptions may be relaxed. The algorithm presented below may be modified to work for uncontrollable systems as long as the uncontrollable eigenvalues are included in Λ. Also, closed-loop poles may have multiplicities up to the number of plant inputs. The details of these modifications are omitted due to lack of space.

The PBH controllability test [1] says that, for controllable systems, the $n \times (n+p)$ matrices

$$\mathbf{P}(\lambda_i) = \left[\, (\lambda_i \mathbf{I} - \mathbf{A}) \quad \mathbf{B} \,\right], \; i = 1, \cdots, n \tag{1}$$

have rank n, and thus have p-dimensional null spaces. Let the $(n+p) \times p$ matrices

$$\begin{bmatrix} \mathbf{U}_i \\ \mathbf{V}_i \end{bmatrix}, \; i = 1, \cdots, n \tag{2}$$

contain orthonormal bases for the null spaces of $\mathbf{P}(\lambda_i)$, where the $\mathbf{U}_i$ matrices contain n rows and the $\mathbf{V}_i$ matrices contain p rows. These null-space basis matrices may be computed using a singular value decomposition of the matrices $\mathbf{P}(\lambda_i)$.

In order for λ_i to be an eigenvalue of $\mathbf{A}-\mathbf{B}\mathbf{K}$, there must be an associated eigenvector $\mathbf{u}_i$ satisfying

$$(\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{u}_i = \lambda_i \mathbf{u}_i, \; i = 1, \cdots, n, \tag{3}$$

which may be rewritten as

$$\left[\, (\lambda_i \mathbf{I} - \mathbf{A}) \quad \mathbf{B} \,\right] \begin{bmatrix} \mathbf{u}_i \\ \mathbf{K}\mathbf{u}_i \end{bmatrix} = \mathbf{0}, \; i = 1, \cdots, n. \tag{4}$$

Note that the eigenvector $\mathbf{u}_i$ may be multiplied by a nonzero scale factor without changing the validity of this equation. This equation shows that

$$\begin{bmatrix} \mathbf{u}_i \\ \mathbf{K}\mathbf{u}_i \end{bmatrix} \in \text{null}(\mathbf{P}(\lambda_i)), \; i = 1, \cdots, n \tag{5}$$

and these null-space vectors can be written in terms of the basis vectors given in (2) as follows

$$\begin{bmatrix} \mathbf{u}_i \\ \mathbf{K}\mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{U}_i \\ \mathbf{V}_i \end{bmatrix} \boldsymbol{\alpha}_i = \begin{bmatrix} \mathbf{U}_i\boldsymbol{\alpha}_i \\ \mathbf{V}_i\boldsymbol{\alpha}_i \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{t}_i \\ \mathbf{b}_i \end{bmatrix}, \; i = 1, \cdots, n \tag{6}$$

where $\boldsymbol{\alpha}_i$ are $p \times 1$ coefficient vectors.

The leftmost vector in (6) shows that the bottom part of the vector equals $\mathbf{K}$ times the top part, and this relationship also holds for the rightmost vectors in (6) for $i = 1, \cdots, n$. Thus

$$\mathbf{KF} = \mathbf{B}, \text{ where } \mathbf{F} = [\, \mathbf{t}_1 \quad \cdots \quad \mathbf{t}_n \,], \; \mathbf{B} = [\, \mathbf{b}_1 \quad \cdots \quad \mathbf{b}_n \,]. \tag{7}$$

If the $\boldsymbol{\alpha}_i$ vectors are chosen so that $\mathbf{F}$ is a nonsingular matrix, then the feedback gain matrix is obtained as

$$\mathbf{K} = \mathbf{BF}^{-1}. \tag{8}$$

It is always possible to choose $\boldsymbol{\alpha}_i$ to insure the nonsingularity of $\mathbf{F}$ [3].

B. Parameterization of Gain Matrices

From (6) it is clear that a particular choice of coefficient vectors $\{\boldsymbol{\alpha}_i\}$ determines the matrices $\mathbf{F}$ and $\mathbf{B}$, which correspond to the unique gain matrix given by (8). Thus, one might want to use $\{\boldsymbol{\alpha}_i\}$ as a set of parameters to describe all possible gain matrices corresponding to a given set of closed-loop pole locations. However, $\{\boldsymbol{\alpha}_i\}$ is not a minimal parameterization. The reason is as follows. Consider a set of coefficient vectors $\{d_i\boldsymbol{\alpha}_i\}$ where d_i are nonzero real numbers. Using the rightmost equality in (6) along with (8) shows that the gain matrix corresponding to the coefficient vectors $\{d_i\boldsymbol{\alpha}_i\}$ is

$$\mathbf{K} = \mathbf{BD}(\mathbf{FD})^{-1} = \mathbf{BF}^{-1}, \; \mathbf{D} = \text{diag}\{d_i\}, \tag{9}$$

which is the same gain matrix given by $\{\boldsymbol{\alpha}_i\}$. Thus, the gain matrix $\mathbf{K}$ is unchanged by a nonzero scaling of the coefficient vectors $\{\boldsymbol{\alpha}_i\}$. In order to remove this redundancy we constrain the norm of each coefficient vector to be unity. That is, each vector $\boldsymbol{\alpha}_i$ is constrained to be a point on a unit hypersphere. This can be accomplished for real-valued coefficient vectors $\boldsymbol{\alpha} = [\, \alpha(1) \quad \alpha(2) \quad \cdots \quad \alpha(p) \,]^T$ using the hyperspherical coordinate vector $\boldsymbol{\theta} = [\, \theta_1 \quad \cdots \quad \theta_{p-1} \,]$ as follows:

$$\begin{aligned} \alpha(1) &= \cos(\theta_1) \\ \alpha(2) &= \sin(\theta_1)\cos(\theta_2) \\ \alpha(3) &= \sin(\theta_1)\sin(\theta_2)\cos(\theta_3) \\ \vdots \;\; &= \;\; \vdots \\ \alpha(p-1) &= \sin(\theta_1)\cdots\sin(\theta_{p-2})\cos(\theta_{p-1}) \\ \alpha(p) &= \sin(\theta_1)\cdots\sin(\theta_{p-2})\sin(\theta_{p-1}). \end{aligned} \tag{10}$$

The case of complex-valued $\boldsymbol{\alpha}_i$ vectors is dealt with in the next subsection. Note that the $(p-1) \times 1$ vector $\boldsymbol{\theta}$ parameterizes all possible real-valued unit-norm $p \times 1$ vectors $\boldsymbol{\alpha}$. In order to make the parameterization unique, each element of $\{\theta_1, \cdots, \theta_{p-2}\}$ should be in an interval of length π and θ_{p-1} should be in an interval of length 2π. In the optimization approach described below we let the elements of $\boldsymbol{\theta}$ be unconstrained because the intervals the θ parameters end up in are not important as long as they specify the optimal vector $\boldsymbol{\alpha}$.

The constraint on the nonsingularity of $\mathbf{F}$ is built into the cost function, which assigns a large cost to parameter vectors that give rise to a $\mathbf{F}$ with a large condition number.

C. Parameterization for Complex-Valued Poles

Suppose λ_i and λ_{i+1} are complex-conjugate desired closed-loop pole locations; that is, $\lambda_{i+1} = \lambda_i^*$, where the asterisk denotes complex conjugate. In this case the basis vectors for the null space of $\mathbf{P}(\lambda_{i+1})$ can be chosen to be the complex conjugates of the null-space vectors of $\mathbf{P}(\lambda_i)$; that is

$$\begin{bmatrix} \mathbf{U}_{i+1} \\ \mathbf{V}_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_i^* \\ \mathbf{V}_i^* \end{bmatrix}. \tag{11}$$

If $\boldsymbol{\alpha}_i$ is a complex-valued vector of unit norm then we must choose $\boldsymbol{\alpha}_{i+1}$ to be the complex conjugate of $\boldsymbol{\alpha}_i$ in order to satisfy the requirement that the eigenvectors $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ be complex conjugates (see (6)). With $\boldsymbol{\alpha}_{i+1} = \boldsymbol{\alpha}_i^*$, (6) gives

$$\begin{bmatrix} \mathbf{t}_{i+1} \\ \mathbf{b}_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_i^* \\ \mathbf{b}_i^* \end{bmatrix} \tag{12}$$

and (7) becomes

$$\mathbf{K}[\cdots \quad \mathbf{t}_i \quad \mathbf{t}_i^* \quad \cdots] = [\cdots \quad \mathbf{b}_i \quad \mathbf{b}_i^* \quad \cdots]. \tag{13}$$

This equation can be multiplied on the right by an identity matrix with row and column i and $i+1$ replaced by:

$$\begin{bmatrix} 1 & j \\ 1 & -j \end{bmatrix}. \tag{14}$$

The result of this multiplication is

$$\mathbf{K}[\cdots \quad \text{re}(\mathbf{t}_i) \quad \text{im}(\mathbf{t}_i) \quad \cdots] = [\cdots \quad \text{re}(\mathbf{b}_i) \quad \text{im}(\mathbf{b}_i) \quad \cdots]. \tag{15}$$

If the procedure described above is performed for every pair of complex-conjugate desired closed-loop pole locations then an equation similar to (15) is obtained, which is a real-valued equation for the gain matrix $\mathbf{K}$.

The parameterization of a complex-valued unit norm vector $\boldsymbol{\alpha}$ can be accomplished by choosing

$$\boldsymbol{\alpha} = \begin{bmatrix} \rho_1 \\ \rho_2 e^{j\phi_1} \\ \rho_3 e^{j\phi_2} \\ \vdots \\ \rho_p e^{j\phi_{p-1}} \end{bmatrix}, \tag{16}$$

where $\boldsymbol{\rho} = [\, \rho_1 \quad \cdots \quad \rho_p \,]^T$ is a real-valued vector of unit norm and the phase angles $\phi_1, \cdots, \phi_{p-1}$ each lie in an interval of length 2π. Notice that the first element of $\boldsymbol{\alpha}$ has been chosen to be real valued. The reason is that any vector that is complex valued and has unit norm may be multiplied by a unit-magnitude complex number without changing these properties. The phase angle of the complex number may be chosen, without loss of generality, to give a zero phase angle to the first element of the scaled complex vector.

The real-valued, unit-norm vector $\boldsymbol{\rho}$ in (16) may be parameterized by $p-1$ hyperspherical angles as shown in (10). In addition to these parameters, the complex vector $\boldsymbol{\alpha}$ in (16) is parameterized by $p-1$ angles $\phi_1, \cdots, \phi_{p-1}$. The result is that complex-conjugate desired closed-loop pole locations require $2(p-1)$ parameters when forming (15) for the gain matrix $\mathbf{K}$. Each real-valued desired closed-loop pole location requires $p-1$ hyperspherical angles. Thus, there are a total of $n(p-1)$ parameters need to parameterize all gain matrices corresponding to a given set of distinct closed-loop pole locations.

D. Optimization

In the previous subsection it was shown that the parameterization of feedback gain matrices corresponding to a given set of desired closed-loop pole locations requires $p-1$ hyperspherical angles for each real valued pole, while a pair of complex-conjugate poles requires $p-1$ hyperspherical angles and $p-1$ phase angles. Let the complete set of angles be placed in a parameter vector $\mathbf{p}$. This vector contains all of the information needed to generate the $\boldsymbol{\alpha}_i$ vectors for all of the desired closed-loop poles using (10) and/or (16). The corresponding gain matrix is then computed from (6)-(8).

We seek the vector $\mathbf{p}^*$ that minimizes the cost function $\boldsymbol{\mathcal{C}}(\mathbf{p})$:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \boldsymbol{\mathcal{C}}(\mathbf{p}) \tag{17}$$

The cost functions of interest are H_∞ robustness norms, which are neither convex nor unimodal. Thus, care must be taken in attempting to minimize $\boldsymbol{\mathcal{C}}(\mathbf{p})$. The approach taken here is to use Matlab's `fminsearch` function, which implements the Nelder-Meade algorithm for nonlinear optimization. Because Nelder-Meade is not a gradient-based algorithm, iteration progress frequently stalls and, following standard practice [10], we reinitialize the iteration. We found it very effective in our tests to perform up to eight restarts The result of the final search is used to construct the optimal gain matrix. One advantage of using this algorithm is that it can avoid some local minima to find a better minimum.

The procedure just described is simple and effective, and was used to generate every example in this paper. However, this approach is simply a "first cut." The favorable results obtained in the examples in Section III serve as motivation to explore a more rigorous approach to the optimization of these cost functions.

E. Parameterizing Observer Gain Matrices

Using the duality between observer and feedback gains, the procedure for parameterizing feedback gain matrices may be used to parameterize observer gain matrices. For a plant with m outputs, each real-valued desired observer pole requires $m-1$ hyperspherical angles, while each complex conjugate pair of poles requires $m-1$ hyperspherical angles and $m-1$ phase angles. Thus, there are a total of $n(m-1)$ parameters needed to parameterize all observer gain matrices that yield a specified set of observer pole locations.

II. COST FUNCTIONS AND MATLAB CODE

This section presents four types of state-space control systems and the robustness norms that will be used as cost functions for the optimization approach to gain matrix calculations. The four control systems are: state-feedback regulators, observer-based regulators, state-feedback tracking systems, and observer-based tracking systems.

A. State Feedback Regulators

Consider first the state-feedback regulator with input-multiplicative plant perturbation shown in Fig. 1. If the

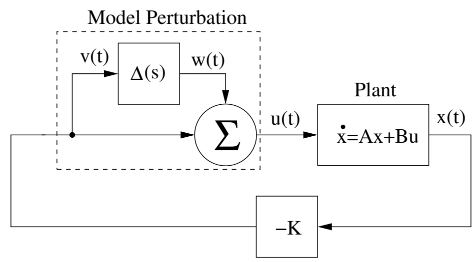

Fig. 1. State-feedback regulator with input-multiplicative plant model perturbation.

nominal closed-loop system (with $\boldsymbol{\Delta}(s) = 0$) is stable, then the closed-loop system remains stable for all model perturbations satisfying

$$\|\boldsymbol{\Delta}(s)\|_\infty < \delta_1, \tag{18}$$

where the system infinity norm is

$$\|\boldsymbol{\Delta}(s)\|_\infty \stackrel{\text{def}}{=} \sup_\omega \bar{\sigma}(\boldsymbol{\Delta}(j\omega)), \tag{19}$$

$$\bar{\sigma}(\mathbf{M}) = \text{ maximum singular value of } \mathbf{M},$$

and δ_1 is the reciprocal of the infinity norm of the system from $\mathbf{w}(t)$ to $\mathbf{v}(t)$. This is simply a statement of the small-gain theorem [5,8], which says that a sufficient condition for the stability of the feedback interconnection of two stable systems is that the product of their system infinity norms be less than unity. From Fig. 1, the system from $\mathbf{w}(t)$ to $\mathbf{v}(t)$, call it $\mathbf{G}_1(s)$, has the following state-space description:

$$\mathbf{G}_1(s) \begin{cases} \dot{\mathbf{x}} = (\mathbf{A}-\mathbf{BK})\mathbf{x} + \mathbf{Bw} \\ \mathbf{v} = -\mathbf{Kx} \end{cases} \tag{20}$$

Then

$$\delta_1 = \frac{1}{\|\mathbf{G}_1(s)\|_\infty} \tag{21}$$

is the stability robustness bound shown in (18).

To calculate $\mathbf{K}$ using pole placement, a set of closed-loop poles must be chosen. Suppose the variable `spoles` contains the desired closed-loop poles. Then the feedback gain matrix may be computed using Matlab's `place` command:

$$\mathbf{K} = \texttt{place}(\mathbf{A}, \mathbf{B}, \texttt{spoles}).$$

The robustness bound δ_1 corresponding to this gain matrix may be calculated using (20) and (21). The value of δ_1

is not guaranteed to be larger than any nonzero number. The `place` function is based on an algorithm that attempts to make the eigenvectors of $\mathbf{A}-\mathbf{BK}$ "as orthogonal as possible" [4]. This is not directly related to optimizing the robustness bound in (21). A Matlab function `rfbg` (robust feedback gains) has been written to implement the optimization approach presented in this paper. It is called in a manner similar to `place`, but with an additional argument for sampling interval, which is defined to be zero for continuous time systems. The reason is that Matlab uses the sampling interval when computing the system infinity norm for discrete-time systems. The call to `rfbg` for continuous-time systems is

$$\mathbf{K} = \mathtt{rfbg}(\mathbf{A}, \mathbf{B}, \mathtt{spoles}, 0).$$

The feedback gains computed by this function give a regulator with the best possible robustness bound for the given set of poles.

This leads to the question, how do we know that the desired closed-loop poles allow for a good robustness bound? As pointed out in [9], "The closed-loop poles can be assigned to arbitrary locations if the system is observable and reachable. However, if we want to have a robust closed-loop system, the poles and zeros of the process [plant] impose severe restrictions on the location of the closed loop poles." A suggested set of rules for choosing closed-loop pole locations that satisfy a settling-time requirement and yield a good robustness bound is given in Table I.

B. Observer-Based Regulator

Consider the observer-based regulator shown in Fig. 2. To

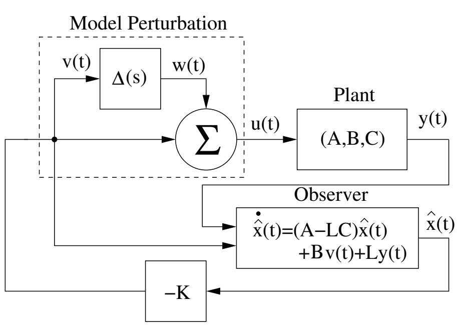

Fig. 2. Observer-based regulator with input-multiplicative plant model perturbation.

compute the robustness bound for this control system, we need a description of the system from $\mathbf{w}(t)$ to $\mathbf{v}(t)$, call it $\mathbf{G}_2(s)$:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\hat{\mathbf{x}}} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & (\mathbf{A}-\mathbf{LC}-\mathbf{BK}) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{w} \\ \mathbf{v} &= \begin{bmatrix} \mathbf{0} & -\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix} \end{aligned} \tag{22}$$

Then

$$\delta_2 = \frac{1}{\|\mathbf{G}_2(s)\|_\infty} \tag{23}$$

is the stability robustness bound for an observer-based regulator. From (22) and (23) it is clear that δ_2 depends on both $\mathbf{K}$ and $\mathbf{L}$. For a MIMO system with p inputs and m outputs, the $\mathbf{K}$ and $\mathbf{L}$ matrices corresponding to some desired closed-loop poles and observer poles are parameterized by $(p-1)n$ and $(m-1)n$ parameters, respectively. One could optimize the robustness bound δ_2 jointly over all the parameters. However, preliminary results indicate that a sequential optimization of δ_2 over $\mathbf{K}$ and $\mathbf{L}$ gives good results. That is, $\mathbf{K}$ is calculated first by optimizing δ_1, the robustness bound for a state-feedback regulator. With the value of $\mathbf{K}$ fixed, δ_2 is then optimized over the parameterization of $\mathbf{L}$. Suppose the variable `spoles` contains the desired closed-loop pole locations and the variable `opoles` contains the desired observer pole locations. The calculation of $\mathbf{K}$ and $\mathbf{L}$ for an observer-based regulator may then be accomplished in Matlab using the following code:

$$\begin{aligned} \mathbf{K} &= \mathtt{rfbg}(\mathbf{A}, \mathbf{B}, \mathtt{spoles}) \\ \mathbf{L} &= \mathtt{obg\_reg}(\mathbf{A}, \mathbf{B}, \mathbf{K}, \mathtt{opoles}) \end{aligned}$$

where `obg_reg` is a new Matlab function that calculates observer gains by optimizing the robustness norm δ_2 for a given $\mathbf{K}$. Note that in the traditional way of computing observer gains

$$\mathbf{L} = \mathtt{place}(\mathbf{A}', \mathbf{C}', \mathtt{opoles})',$$

the value of the feedback gain matrix $\mathbf{K}$ is not used. The traditional approach is based on the separation principle [1] and the duality between closed-loop pole placement and observer-pole placement. In the proposed approach the observer gains are not calculated separately from the feedback gains. In order to obtain an observer-based regulator with good robustness, the calculation of $\mathbf{L}$ must include all of the information needed to evaluate the robustness bound δ_2.

C. State-Feedback Tracking System

Consider the state-feedback tracking system shown in Fig. 3. Before presenting the stability robustness bound

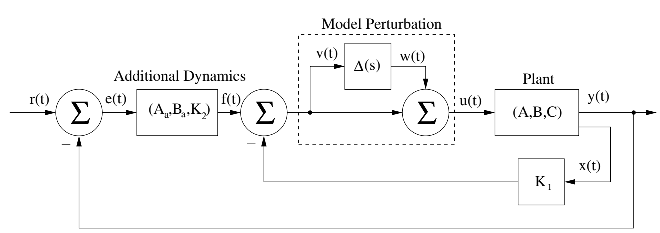

Fig. 3. State-feedback tracking system with input-multiplicative plant model perturbation.

for this control system, we first mention that the design of a tracking system begins with the specification of the additional dynamics [2],

$$\dot{\mathbf{x}}_a = \mathbf{A}_a \mathbf{x}_a + \mathbf{B}_a \mathbf{e}. \tag{24}$$

For example, to eliminate steady-state errors for step inputs and constant disturbances the additional dynamics system is chosen to be a parallel set of integrators. In this case, for a p-input, p-output plant, the $p \times p$ additional dynamics matrices are

$$\mathbf{A}_a = \mathbf{0}, \quad \mathbf{B}_a = \mathbf{I}. \tag{25}$$

In order to compute the $p \times n$ state-feedback gain matrix $\mathbf{K}_1$ and the $p \times p$ additional dynamics gain matrix $\mathbf{K}_2$, a design model consisting of the open-loop cascade of the plant followed by the additional dynamics is needed [2]. This design model is given by:

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B}_a\mathbf{C} & \mathbf{A}_a \end{bmatrix}, \ \mathbf{B}_d = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}. \tag{26}$$

Given a desired set of $n+p$ closed-loop poles, call it `spoles`, a $p \times (n+p)$ gain matrix $\mathbf{K}_d$ is computed by pole placement; e.g. using the Matlab `place` command or the new function `rfbg`. Then $\mathbf{K}_1$ consists of the first n columns of $\mathbf{K}_d$ and $\mathbf{K}_2$ is the last p columns of $\mathbf{K}_d$. To summarize, the calculation of gains for a state-feedback tracking system is accomplished by calculating a gain matrix $\mathbf{K}_d$ that regulates the design model $(\mathbf{A}_d, \mathbf{B}_d)$. It may be shown that the stability robustness bound for a state-feedback tracking system in Fig. 3 is simply the bound for a state-feedback regulator given by (20) and (21) with $(\mathbf{A}, \mathbf{B}, \mathbf{K})$ replaced by $(\mathbf{A}_d, \mathbf{B}_d, \mathbf{K}_d)$. To summarize, once the design model is formed using (26), either `place` or `rfbg` may be used to calculate $\mathbf{K}_d$, which is then separated into $\mathbf{K}_1$ and $\mathbf{K}_2$.

D. Observer-Based Tracking System

Consider the observer-based tracking system shown in Fig. 4. Using the state vector $[\,\mathbf{x}^T \quad \hat{\mathbf{x}}^T \quad \mathbf{x}_a^T\,]^T$, the system from $\mathbf{w}(t)$ to $\mathbf{v}(t)$, call it $\mathbf{G}_3(s)$, is described by the following state-space matrices:

$$\begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{K}_1 & \mathbf{B}\mathbf{K}_2 \\ \mathbf{L}\mathbf{C} & (\mathbf{A}-\mathbf{L}\mathbf{C}-\mathbf{B}\mathbf{K}_1) & \mathbf{B}\mathbf{K}_2 \\ -\mathbf{B}_a\mathbf{C} & \mathbf{0} & \mathbf{A}_a \end{bmatrix}, \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{27}$$
$$[\,\mathbf{0} \quad -\mathbf{K}_1 \quad \mathbf{K}_2\,].$$

The stability robustness bound for the observer-based tracking system is

$$\delta_3 = \frac{1}{\|\mathbf{G}_3(s)\|_\infty}. \tag{28}$$

As with the observer-based regulator, we use a sequential optimization by first optimizing δ_1 with respect to the feedback gain matrix $\mathbf{K}_d = [\,\mathbf{K}_1 \quad \mathbf{K}_2\,]$ and then optimizing δ_3 with respect to the observer gain matrix $\mathbf{L}$. This is accomplished using (25), (26) and the following Matlab commands:

$$\begin{aligned} &\mathtt{Kd} = \mathtt{rfbg}(\mathtt{Ad}, \mathtt{Bd}, \mathtt{spoles}, \mathtt{T}) \\ &\mathtt{K1} = \mathtt{Kd}(:, 1:\mathtt{n}) \\ &\mathtt{K2} = \mathtt{Kd}(:, \mathtt{n}+1:\mathtt{n}+\mathtt{p}) \\ &\ \mathtt{L} = \mathtt{obg\_ts}(\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{Aa}, \mathtt{Ba}, \mathtt{K1}, \mathtt{K2}, \mathtt{opoles}, 0) \end{aligned} \tag{29}$$

where `obg_ts` is a new Matlab function that calculates observer gains by optimizing δ_3 for fixed values of $\mathbf{K}_1$ and $\mathbf{K}_2$. Notice that in the proposed approach, an observer designed for a given plant $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ will have an observer gain matrix that depends on the specific control system, e.g. for an observer-based regulator $\mathbf{L}$ is calculated to minimize δ_2, while for an observer-based tracking system, $\mathbf{L}$ is calculated to minimize δ_3. The traditional approach based on the separation principle gives only one observer gain matrix, regardless how the observer is used.

III. EXAMPLES

A. Comparison with Linear Quadratic Regulators

Given a controllable p-input system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ that is regulated by state feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$, the feedback gain matrix $\mathbf{K}$ that minimizes the quadratic cost function

$$J = \int_0^\infty \left[\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\right] dt \tag{30}$$

may be computed by the Matlab function

$$\mathbf{K} = \mathtt{lqr}(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}), \tag{31}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are positive definite matrices. The LQR gain matrix has the desirable property that the resulting closed-loop system has an input-multiplicative stability robustness norm that is guaranteed to be greater than or equal to 0.5 [8]. There is no such robustness guarantee for state-feedback regulators when the gain matrix $\mathbf{K}$ is computed by standard pole-placement methods such as Matlab's `place` command.

The purpose of this first example is to show that when gain matrices are computed using the proposed `rfbg` algorithm the resulting stability robustness norm is at least as good as that of LQR, and it is usually better. The systems in this example are 100 randomly generated 3-input, 7th-order systems. The matrices $\mathbf{A}$ and $\mathbf{B}$ for each system consist of i.i.d. random variables with zero mean and unit variance. For each system, the LQR gain matrix was computed using identity weighting matrices:

$$\mathtt{K1} = \mathtt{lqr}(\mathtt{A}, \mathtt{B}, \mathtt{eye}(7), \mathtt{eye}(3)).$$

Then the closed-loop pole locations were computed and used in the pole-placement algorithms `place` and `rfbg` to calculate two additional gain matrices

$$\begin{aligned} \mathtt{spoles} &= \mathtt{eig}(\mathtt{A} - \mathtt{B} * \mathtt{K1}) \\ \mathtt{K2} &= \mathtt{place}(\mathtt{A}, \mathtt{B}, \mathtt{spoles}) \\ \mathtt{K3} &= \mathtt{rfbg}(\mathtt{A}, \mathtt{B}, \mathtt{spoles}, 0). \end{aligned} \tag{32}$$

By construction, all three gain matrices result in the identical closed-loop pole locations. For each trial system, the stability robustness bound δ_1 was calculated for the regulator gain matrices calculated by three different algorithms: LQR, place, and rfbg. The results are shown in Fig. 5.

B. Observer-Based Regulator for Cart-Pendulum System

Consider the following state-space model $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ for a cart-pendulum system, linearized about the inverted pendu-

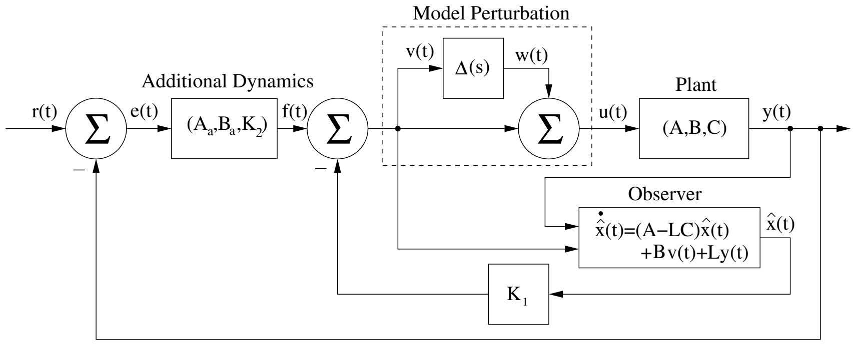

Fig. 4. Observer-based tracking system with input-multiplicative plant model perturbation.

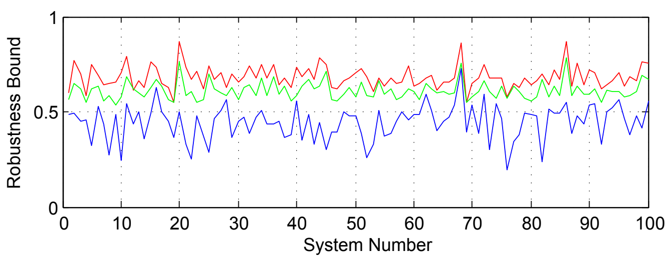

Fig. 5. Robustness bounds for state-feedback regulators designed by three different methods for 100 randomly generated systems . The regulators for the blue (bottom) curve were designed using `place`. The green (middle) curve is for LQR regulators. The red (upper) curve is for regulators designed using `rfbg`.

lum position, described in [2]:

$$\mathbf{A} = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ 23.1000 & 0 & 0 & 0.1189 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 0 & 0 & -25.0000 \end{bmatrix}, \qquad (33)$$
$$\mathbf{B} = \begin{bmatrix} 0 \\ 12.525 \\ 0 \\ 2633 \end{bmatrix}, \; \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The state variables are x_1=pendulum position (rad), x_2=pendulum velocity (rad/sec), x_3=motor position (rad), and x_4=motor velocity (rad/sec). The motor shaft is directly connected to a 495 rad/m lead screw. The plant input $u(t)$ is the voltage applied to the motor power amplifier. This state-space model describes a hardware system built at the University of Rhode Island.

The desired settling time for the closed-loop system is $T_s = 1$ sec. In order to achieve this, the desired closed-loop poles are chosen to be the roots of a 4th-order normalized Bessel polynomial (see Table II). Because this is a single-input system, the calculation of feedback gains via pole placement has a unique result, which can be obtained using the Matlab `place` function. The result is

$$\mathbf{K} = [\,1.7032 \quad 0.3609 \quad -0.0229 \quad -0.0040\,]. \qquad (34)$$

The stability robustness norm δ_1 for this regulator, as well as classical stability margins, are given in the first row of Table III. Although this control system has a nice transient response, its robustness norm and stability margins are inadequate. These could be improved by choosing a different set of desired closed-loop pole locations, e.g. using regulator rule 2 from Table I. For the purpose of this example, however, we keep these feedback gains and calculate observer gains for an observer-based regulator. In the calculations that follow, we use 4th-order Bessel poles, scaled to have a settling time that is a quarter of the regulator settling time, namely $T_s/4 = 0.25$ sec. That is,

$$\mathtt{opoles} = \mathtt{s4}/(\mathtt{Ts}/4).$$

The observer gain matrix could be calculated by `L=place(A',C',opoles)'`. The result is

$$\mathbf{L}_{\mathtt{place}} = \begin{bmatrix} 33.0996 & 1.9076 \\ 664.4218 & 72.2354 \\ -16.3186 & 18.2495 \\ 81.9758 & 62.9032 \end{bmatrix}. \qquad (35)$$

The stability robustness norm δ_2 for this observer-based regulator system, as well as classical stability margins, are given in the second row of Table III. It can be seen that this observer-based regulator has much worse stability robustness than the state feedback regulator. However, this degradation of stability robustness is not a necessary consequence of using an observer. Because the cart/pendulum system has two measured variables, the full-order observer gain matrix corresponding to some choice of observer poles is not unique. The `place` algorithm chooses the particular $\mathbf{L}$ matrix that makes the eigenvectors of $(\mathbf{A} - \mathbf{LC})$ as orthogonal as possible. However, this orthogonality has no direct relationship to the stability robustness of the observer-based regulator. This robustness depends on the system infinity norm of the system shown in (27). Note that this system is a function of the feedback gain vector $\mathbf{K}$. With $\mathbf{K}$ given by (34), the cost function depends on the unknown observer gain matrix, which can be calculated using `obg_reg` to maximize the robustness norm. The result is

$$\mathbf{L}_{\mathtt{obg\_reg}} = \begin{bmatrix} 32.0684 & 1.0971 \\ 694.1691 & 26.1277 \\ 1.6830 & 19.2807 \\ 16.3545 & 51.8820 \end{bmatrix}. \qquad (36)$$

The rules given below for choosing desired closed-loop pole or observer-pole locations are for continuous-time systems. For digital control (discrete-time systems), these numbers, p_i, must be mapped using the ZOH pole-mapping formula, $e^{p_i T}$, where T is the sampling interval in seconds.

Regulator Poles

1) Use normalized Bessel poles scaled (divided) by the desired settling time (see Table II).
2) Use sufficiently damped plant poles; that is, plant poles whose real parts lie to the left of `s1/Ts`.
3) If the plant has complex poles that are not sufficiently damped, choose closed-loop poles by replacing the real parts of these poles with `s1/Ts` and keeping the imaginary parts the same. Changing the real parts adds damping to those plant poles, and so these are called "added damping" poles.
4) If the plant has unstable pole locations (real part greater than zero) consider the reflection of these poles about the imaginary axis, i.e. change the sign of the (real-part) of the pole. If the reflected pole is to the left of `s1/Ts` use it as a closed-loop pole.

Tracking System Poles

1) Use all of the rules for regulator poles
2) If the plant has "slow," stable zeros, consider using these as closed-loop poles for the tracking system. A slow, stable zero is a zero whose real part is negative and to the right of `4*s1/Ts`. Note that this slow closed-loop pole will not affect the settling time of the tracking system because it is canceled by the corresponding plant zero.

Observer Poles

1) Use normalized Bessel poles scaled (divided) by the desired observer settling time. The observer settling time should be chosen to be one third to one fifth of the desired settling time for the regulator or tracking system. Such an observer is said to be "three to five times faster."
2) If the plant has stable zeros, consider using these zero locations as observer poles.
3) If the plant has zeros in the right-half plane, the reflection of these zero locations (replace their real parts with the negatives of their real parts) should be chosen as observer poles.

TABLE I

RULES FOR SELECTING POLE LOCATIONS

The stability robustness norm δ_2 for this observer-based regulator system, as well as classical stability margins, are given in the third row of Table III. It can be seen that this observer-based regulator has very good stability robustness, even much better than the state feedback regulator. This example shows that an observer-based regulator is not limited to a "recovery" of state-feedback robustness; it can greatly exceed it.

C. Lateral Control of an F-16 Aircraft

This example is taken from [6,7], where an LQG/LTR controller is designed. The model for the F-16 aircraft is linearized at nominal flight conditions, and includes actuator dynamics for the aileron and rudder. The state-space model is given as follows. The matrix $\mathbf{A}$ is

$$\begin{bmatrix} -0.322 & 0.0640 & 0.0364 & -0.9917 & 0.0003 & 0.0008 \\ 0 & 0 & 1 & 0.0037 & 0 & 0 \\ -30.6492 & 0 & -3.6784 & 0.6646 & -0.7333 & 0.1315 \\ 8.53950 & -0.0254 & -0.4764 & -0.0319 & -0.0620 & \\ 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 \end{bmatrix}$$

Variable Name	Pole Locations
$\mathbf{s}_1$	-4.6200
$\mathbf{s}_2$	$-4.0530 \pm j2.3400$
$\mathbf{s}_3$	$-5.0093,\ -3.9668 \pm j3.7845$
$\mathbf{s}_4$	$-4.0156 \pm 5.0723,\ -5.5281 \pm j1.6553$

TABLE II

NORMALIZED BESSEL POLES FOR 1ST THROUGH 4TH-ORDER SYSTEMS WITH 1-SECOND SETTLING TIME (FROM [2]). TO GET A SETTLING TIME OF T_S SECONDS, DIVIDE ALL POLES BY T_S.

Regulator	δ	UGM	LGM	PM
$\mathbf{K}$	0.30	10.0	-4.2	20
$\mathbf{K},\mathbf{L}_{\texttt{place}}$	0.10	2.6	-0.9	7
$\mathbf{K},\mathbf{L}_{\texttt{obg\_reg}}$	0.60	9.8	-8.9	36

TABLE III

ROBUSTNESS BOUND δ AND CLASSICAL STABILITY MARGINS, UPPER/LOWER GAIN MARGINS (UGM,LGM) AND PHASE MARGIN (PM), FOR CART/PENDULUM SYSTEM.

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 20.2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & 57.2958 & 0 & 0 & 0 & 0 \\ 57.2958 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The inputs and outputs are

$$\begin{aligned} u_1 &= \text{commanded aileron angle (rad)} \\ u_2 &= \text{commanded rudder angle(rad)} \\ y_1 &= \text{roll angle (degrees)} \\ y_2 &= \text{lateral side slip angle (deg)} \end{aligned}$$

The plant poles are $-20.2,\ -20.2,\ -3.6152.\ -0.4226 \pm j3.0634,\ -0.0163$. The plant has a single zero at -79.15. The plant zero is so far into the left-half plane that it does not influence the choice of closed-loop poles. However, the locations of the plant poles triggers several of the rules given in Table I.

To get a desired settling time of 1.25 seconds we choose $T_s = 1.25$. Then the first two plant poles shown above are to the left of `s1/Ts` (see Table II). These two sufficiently damped poles will be kept as desired closed-loop poles, although they will be split into two distinct poles. The complex-conjugate eigenvalues are not sufficiently damped. The corresponding closed-loop poles will replace the real part of -0.4226 with `s1/Ts`=-3.7. The remaining closed-loop poles are chosen as scaled 4th-order Bessel poles. Thus, the desired closed-loop poles are chosen as follows:

$$\begin{aligned} \texttt{cp} &= -20.2 + .1 * \texttt{j} \\ \texttt{adp} &= \texttt{s1/Ts} + \texttt{j} * 3.0634 \\ \texttt{spoles} &= [\texttt{cp conj(cp) adp conj(adp) s4/Ts}] \end{aligned}$$

The observer settling time is chosen to be `Tso=Ts/3` for a 3-times faster observer. The desired observer poles are then chosen as scaled 6th-order Bessel poles (from [2]): `opoles = s6/Tso`. With the additional dynamics chosen to be two integrators in parallel, (25) and (26) and the code shown in (29) yields the following gain matrices:

$$\mathbf{K}_1 = \begin{bmatrix} 579.5 & -164.8 & -19.44 & -107.3 & .7025 & 0.7159 \\ 378.5 & -29.598 & -0.3156 & -108.0 & 0.1500 & 0.1978 \end{bmatrix}$$

$$\mathbf{K}_2 = \begin{bmatrix} -7.1790 & 22.0422 \\ -2.1696 & 17.9255 \end{bmatrix}, \; \mathbf{L} = \begin{bmatrix} -0.2154 & 0.1767 \\ 0.5144 & 0.7804 \\ 1.561 & 6.556 \\ -0.5141 & -0.8267 \\ -167.06 & 71.94 \\ -780.1 & 400.0 \end{bmatrix}$$

The robustness norm δ_1 for the state-feedback tracking system using $\mathbf{K}_1$ and $\mathbf{K}_2$ is 0.82. The robustness norm δ_3 for the observer-based tracking system is 0.69. This observer-based system has excellent robustness and excellent transient response to a unit step command in roll angle, as shown in Fig. 6. The observer gains could also be calculated with Matlab's `place` command (see line above (35)) using the same observer poles, but the robustness norm for the resulting observer-based tracking system is 0.32, which is poor.

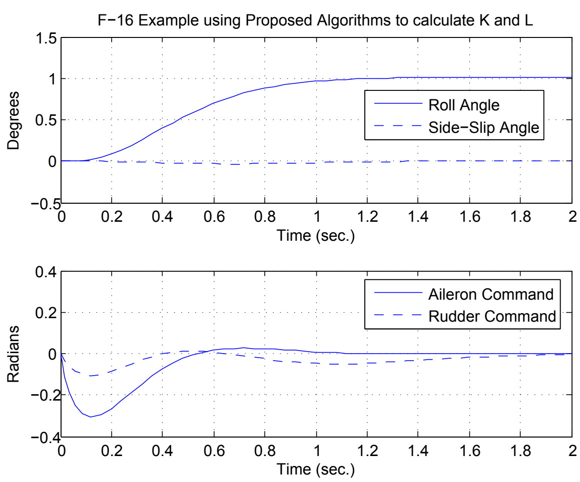

Fig. 6. Simulation results for an observer-based tracking system designed using the proposed `rfbg` and `obt_ts` algorithms.

An LQG/LTR tracking system for this F-16 example was designed in [6,7]. The state-feedback LQR system had an output-multiplicative stability robustness bound of 0.82. The robustness bound of the LQG/LTR observer-based tracking system was also 0.82. However, the LTR procedure that recovers the robustness bound of the LQR system is a limiting procedure that sends some poles to minus infinity [5]. Thus, the transient response may be very poor, as it is in this case. The response to a unit step command in roll angle for the LQG/LTR system designed in [6,7] is shown in Fig. 7. It is completely unacceptable as the maximum rudder command is 100 rads.

IV. Conclusions

The ability to place closed-loop and observer poles enables a designer to effectively address settling time and bandwidth constraints. However, lack of uniqueness of gain matrices

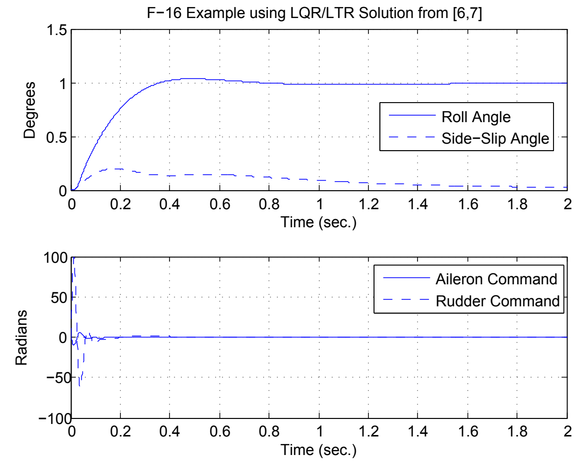

Fig. 7. Simulation results for LQG/LTR tracking system given in [6,7]. The maximum rudder command is 100 rads, which makes this design useless.

for multivariable systems requires choosing a specific gain matrix out of an infinite set of gain matrices, all of which give the specified pole locations. In this paper, an algorithm is given for finding the gain matrix that yields the maximum stability robustness bound. The performance of the proposed algorithm for computing both feedback and observer gains was demonstrated with a number of examples. The ability to calculate pole-placement observer gains to maximize robustness is significant, as traditional approaches to observer gain calculation often yield unacceptable robustness. The Matlab functions `rfbg.m`, `obg_reg.m`, and `obg_ts.m` may be obtained by sending an e-mail to the author.

Acknowledgment

The author would like to thank Prof. Orlando Merino for helpful discussions related to this work.

References

[1] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ; 1980.
[2] R.J. Vaccaro, *Digital Control: A State-Space Approach*, McGraw-Hill, New York, NY; 1995.
[3] B.C. Moore, "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment," *IEEE Trans. Automatic Control*, vol. 21, no. 5, pp. 689–692, October, 1976.
[4] J. Kautsky, N. K. Nichols and P. Van Dooren, "Robust Pole Assignment in Linear State Feedback, *International Journal of Control*, vol. 41, pp. 1129-1155, 1985.
[5] J.B. Burl, *Linear Optimal Control*, Addison Wesley Longman, Menlo Park, CA, 1999.
[6] F.L. Lewis, D.L. Vrabie, and V.L. Syrmos, *Optimal Control*, 3rd edition, John Wiley & Sons, , Hoboken, NJ, 2012.
[7] B.L. Stephens and F.L. Lewis, *Aircraft Control and Simulation*, 2nd edition, John Wiley & Sons, , Hoboken, NJ, 2003.
[8] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design,* 2nd ed., John Wiley & Sons, Chichester, England, 2005.
[9] K.J. Astrom and R.M. Murray, *Feedback Systems*, pg. 362, Princeton University Press, Princeton, NJ, 2008.
[10] W.H. Press et. al, *Numerical Recipes*, 3rd ed., Cambridge University Press, New York, NY 2007.