JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 23, No. 5, September–October 2000

Extension of l_1-Optimal Robust Tracking to the Multivariable Case

Craig N. Scott* and Lincoln A. Wood†
Royal Melbourne Institute of Technology, Melbourne, Victoria 3001, Australia

Previously, a complete approach to forming controllers for single-input/single-output plants was presented. These controllers were designed to track a specific maneuver in the presence of disturbances, sensor noise, and uncertainty in the plant model. By use of a simplified version of the Youla parameterization, the process was split into two optimizations, referred to as tracking and robustness. The method is extended to plants with any number of inputs or outputs.

Nomenclature

D_C = denominator feedback part of controller, $\hat{D}_C/l$
d = denominator of right coprime factorization of the plant, $\hat{d}/r$
l, r = stable polynomials
N_{C_1} = feedforward part of controller
N_{C_2} = numerator feedback part of controller, $\hat{N}_{C_2}/l$
n = numerator of right coprime factorization of the plant, $\hat{n}/r$
P_0 = linear plant model, nd^{-1}, $\tilde{d}^{-1}\tilde{n}$
R = arbitrary matrix of stable, rational polynomials
$\boldsymbol{u}$ = plant inputs
W_{2-4} = weighting filters on the respective inputs (for multi-input/multi-output plants, these are diagonal matrices of weighting filters)
$\boldsymbol{w}_1$ = command references,

$$w_{1_D}^{-1}\begin{Bmatrix}\hat{w}_1\\ \hat{w}_2\\ \vdots\end{Bmatrix} = w_{1_D}^{-1}\hat{\boldsymbol{w}}$$

$\boldsymbol{w}_2$ = disturbance inputs
$\boldsymbol{w}_3$ = sensor noise inputs
$\boldsymbol{w}_4$ = modeling error inputs
X = matrix of stable, rational polynomials, $\hat{X}/l$
$\tilde{X}$ = matrix of stable, rational polynomials, $\hat{\tilde{X}}/l$
Y = matrix of stable, rational polynomials, $\hat{Y}/l$
$\tilde{Y}$ = matrix of stable, rational polynomials, $\hat{\tilde{Y}}/l$
$\boldsymbol{y}$ = plant outputs
ϕ = tracking error, $\boldsymbol{w}_1 - \boldsymbol{y}$

I. Introduction

A TECHNIQUE was detailed in Ref. 1 that allowed controllers suitable for real-world implementation to be formed in the time domain. When we used a simplified version of the Youla parameterization (see Ref. 2) the task was split into two independent optimizations, tracking and robustness. The latter of these two optimizations shall be referred to as the regulation problem for reasons to be discussed later. In the tracking optimization, the l_1 norm of both the tracking error and the actuator demands were minimized. This kept the tracking error low but also penalized excessive actuator usage. The regulation optimization minimized the l_1 norm of the transfer functions relating the disturbances, sensor noise, and modeling errors to the tracking error and actuator demands. This minimized the worst-case response of each of these transfer functions, making the controller optimally insensitive to these environmental effects.

To make the method easier to follow, only single-input/single-output (SISO) plants were considered in Ref. 1. This paper now extends those derivations such that optimizations can also be performed for multi-input/multi-output (MIMO) plants. The tracking optimization requires only minor modification, but the regulation optimization is substantially more complicated. Similar to the approach taken in Ref. 3, the objectives of both optimizations are no longer l_1 norms, but instead, the l_1 norm of each element of the matrices appearing in the objective is minimized. This is a significant departure from conventional l_1 methods.

A common problem with many approaches based on the Youla parameterization is that the notation easily becomes very cluttered and cumbersome. This is dealt with directly in this paper in an effort to maintain some form of consistency in the derivations. A number of operators are introduced to make the work more compact and easier to follow (see Appendix).

II. General Problem Setup

In forming stabilizing controllers, some general constraints need to be satisfied. These guarantee that the controller will not cause either internal or external instability of the system. Figure 1 shows the block diagram of the system, including a multiplicative uncertainty in the plant model (this format for multiplicative uncertainty was derived in Ref. 1 and applies to the MIMO case also). Because this format uses the Youla parameterization, the full Bezout identity must be satisfied as part of ensuring stability:

$$\begin{bmatrix} Y & X \\ -\tilde{n} & \tilde{d} \end{bmatrix}\begin{bmatrix} d & -\tilde{X} \\ n & \tilde{Y} \end{bmatrix} = I \tag{1}$$

An equivalent way to express this using only the plant and controller coprime factorization terms is straightforward to derive. Using Eq. (1) with Eqs. (2–4) gives the result of Eq. (5):

$$N_{C_2} = X + R\tilde{d} \tag{2}$$

$$D_C = Y - R\tilde{n} \tag{3}$$

$$P_0 = nd^{-1} = \tilde{d}^{-1}\tilde{n} \tag{4}$$

$$\Rightarrow N_{C_2}n + D_C d = I \tag{5}$$

Enforcing Eq. (5) ensures that the controller stabilizes the plant for disturbances and sensor noise, but not necessarily for modeling errors. This must be done by adding another constraint to the problem. In the SISO case, it was stated that this was achieved by enforcing $\|W_4 n N_{C_2}\|_{\text{peak gain}} \leq 1$. The result is similar for the MIMO case, as will now be derived.

Received 12 July 1999; accepted for publication 9 February 2000.

*Doctoral Candidate, Department of Aerospace Engineering, Wackett Aerospace Centre, GPO Box 2476V.

†Adjunct Professor of Aerospace Engineering, Department of Aerospace Engineering, Wackett Aerospace Centre, GPO Box 2476V.

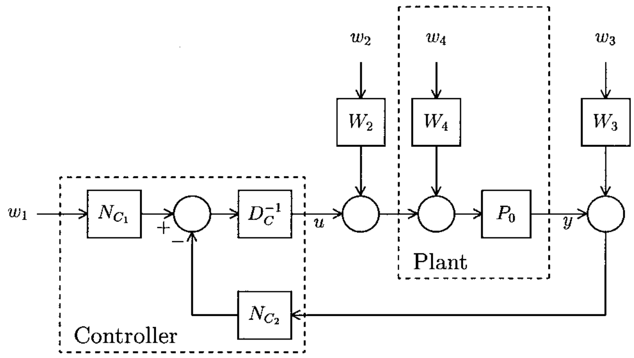


Fig. 1 Two-degree-of-freedom controller.

Because N_{C_1} and N_{C_2} are both stable, these blocks in Fig. 1 will give bounded outputs for bounded inputs. Therefore, to guarantee internal stability for the system, it is sufficient to ensure that the outputs from the controller $\boldsymbol{u}$ and from the plant $\boldsymbol{y}$ are both bounded ($\boldsymbol{w}_{1-4}$ are assumed to be bounded inputs). The following derivation is adapted from Ref. 4.

Consider the controller output, obtained by tracing back along the signal path around the feedback loop:

$$\boldsymbol{u} = D_C^{-1}\left\{N_{C_1}\boldsymbol{w}_1 - N_{C_2}[W_3\boldsymbol{w}_3 + P_0(W_4\boldsymbol{w}_4 + W_2\boldsymbol{w}_2 + \boldsymbol{u})]\right\}$$

$$= D_C^{-1}N_{C_1}\boldsymbol{w}_1 - D_C^{-1}N_{C_2}nd^{-1}W_2\boldsymbol{w}_2 - D_C^{-1}N_{C_2}W_3\boldsymbol{w}_3$$

$$- D_C^{-1}N_{C_2}nd^{-1}W_4\boldsymbol{w}_4 - D_C^{-1}N_{C_2}nd^{-1}\boldsymbol{u} \tag{6}$$

By the use of Eq. (5), the substitution $D_C^{-1}N_{C_2}nd^{-1} = (D_C^{-1}d^{-1} - I)$ can be made that allows the expression to be simplified to

$$\boldsymbol{u} = dN_{C_1}\boldsymbol{w}_1 - (I - dD_C)W_2\boldsymbol{w}_2 - dN_{C_2}W_3\boldsymbol{w}_3 - (I - dD_C)W_4\boldsymbol{w}_4 \tag{7}$$

From Ref. 1, $w_4 = \Delta(u + W_2w_2)$, which is true for the MIMO as well as the SISO case, allowing the following to be written

$$[I + (I - dD_C)W_4\Delta]\boldsymbol{u} = dN_{C_1}\boldsymbol{w}_1$$

$$- (I - dD_C)(W_4\Delta + I)W_2\boldsymbol{w}_2 - dN_{C_2}W_3\boldsymbol{w}_3 \tag{8}$$

By the Youla parameterization, all of the terms on the right-hand side of Eq. (8) are stable, so that if $\boldsymbol{u}$ is to be bounded, $[I + (I - dD_C)W_4\Delta]^{-1}$ must also be stable. When we use the main result of Ref. 5, which relies on the small gain theorem, this is guaranteed if and only if $\|(I - dD_C)W_4\|_1 \leq 1$. If this condition is satisfied, $\boldsymbol{u}$ is bounded and, by consequence of the relation $\boldsymbol{w}_4 = \Delta(\boldsymbol{u} + W_2\boldsymbol{w}_2)$, $\boldsymbol{w}_4$ is also bounded. This will be useful in showing that $\boldsymbol{y}$, too, is therefore bounded. Again, the derivation is begun by tracing back along the signal path, this time from the plant output

$$\boldsymbol{y} = nd^{-1}\left\{W_4\boldsymbol{w}_4 + W_2\boldsymbol{w}_2 + D_C\left[N_{C_1}\boldsymbol{w}_1 - N_{C_2}(W_3\boldsymbol{w}_3 + \boldsymbol{y})\right]\right\}$$

$$\left(I + nd^{-1}D_C^{-1}N_{C_2}\right)\boldsymbol{y} = nd^{-1}D_C^{-1}N_{C_1}\boldsymbol{w}_1 + nd^{-1}W_2\boldsymbol{w}_2$$

$$- nd^{-1}D_C^{-1}N_{C_2}W_3\boldsymbol{w}_3 + nd^{-1}W_4\boldsymbol{w}_4 \tag{9}$$

Premultiplying by $D_C dN_{C_2}$ and using Eq. (5) lead to

$$\boldsymbol{y} = nN_{C_1}\boldsymbol{w}_1 + nD_CW_2\boldsymbol{w}_2 - nN_{C_2}W_3\boldsymbol{w}_3 + nD_CW_4\boldsymbol{w}_4 \tag{10}$$

Because all of the terms on the right-hand side are bounded, it follows that $\boldsymbol{y}$ must be bounded also. Therefore, by enforcing Eq. (5) and the condition $\|(I - dD_C)W_4\|_1 \leq 1$, the controller is guaranteed to stabilize the system. Nevertheless, nothing has yet been said about the sensitivity of the controller to the environmental effects. To minimize the sensitivity to disturbances, etc., use is made of the closed-loop map, a matrix equation relating the reference demands and environmental effects to the actuator demands and tracking error. By our tracing signals around the loop, the MIMO closed-loop map can be written

$$\begin{Bmatrix}\boldsymbol{u}\\ \phi\end{Bmatrix} = \begin{bmatrix} dN_{C_1} & \vdots & (dD_C - I)W_2 & -dN_{C_2}W_3 & (dD_C - I)W_4 \\ I - nN_{C_1} & \vdots & -nD_CW_2 & -dN_{C_2}W_3 & -nD_CW_4 \end{bmatrix}$$

$$\times \begin{Bmatrix}\boldsymbol{w}_1\\ \cdots \\ \boldsymbol{w}_2\\ \boldsymbol{w}_3\\ \boldsymbol{w}_4\end{Bmatrix} \tag{11}$$

$$= \begin{Bmatrix}\boldsymbol{u}^*\\ \phi^*\end{Bmatrix} + \begin{bmatrix} u_2 & u_3 & u_4 \\ \phi_2 & \phi_3 & \phi_4 \end{bmatrix}\begin{Bmatrix}\boldsymbol{w}_2\\ \boldsymbol{w}_3\\ \boldsymbol{w}_4\end{Bmatrix} \tag{12}$$

Our using the Youla parameterization in this way allows the problem to be partitioned as shown, as was done in the SISO case. N_{C_1} and the N_{C_2} and D_C pair can then be determined independently.

III. Tracking

The tracking optimization is concerned with obtaining $\boldsymbol{u}^*$ and ϕ^* (or, equivalently, N_{C_1}). These terms depend only on $\boldsymbol{w}_1$ and represent the optimal actuator demands and tracking error when there are no disturbances, etc. The main differences between the SISO and MIMO derivations are that $\boldsymbol{w}_1$, $\boldsymbol{u}^*$, and ϕ^* become vectors of polynomials (or equivalently, vectors of time sequences) and the polynomial q in the SISO case is replaced by a matrix of polynomials N_{C_1}. The plant's coprime factorization will also be slightly different because n and d cannot just be set to the numerator and denominator of its transfer function when the plant is not SISO (this was equivalent to setting $r = 1$). Methods for determining n and d can be found readily, for example, in Ref. 2.

The steady-state actuator also becomes a vector, but it is related to $\boldsymbol{u}$ in the same way as for the SISO case

$$\boldsymbol{u} = (1 - z^{-1})^{-1}\boldsymbol{u}_{SS} + \bar{\boldsymbol{u}}$$

$$= h^{-1}\boldsymbol{u}_{SS} + \bar{\boldsymbol{u}} \tag{13}$$

With this notation and the $\mathcal{S}$ operator, the constraints for the MIMO problem can be recast from Eq. (11) to the following form:

$$\begin{bmatrix} \left(hrw_{1_D}\right)I & & -\mathcal{S}(\hat{\boldsymbol{w}}, h\hat{d}) \\ & \left(hrw_{1_D}\right)I & \mathcal{S}(\hat{\boldsymbol{w}}, \hat{n}) \end{bmatrix}\begin{Bmatrix}\bar{\boldsymbol{u}}\\ \phi\\ \mathrm{col}\left(N_{C_1}\right)\end{Bmatrix}$$

$$= \begin{Bmatrix} -rw_{1_D}\boldsymbol{u}_{SS} \\ r\hat{\boldsymbol{w}}\end{Bmatrix} \tag{14}$$

The objective for the SISO problem consisted of the sum of the l_1 norms of u and ϕ. In the MIMO problem, the sum of the l_1 norms of each element of u and ϕ is taken. This ensures that the optimization does not just minimize the worst channel, but instead considers all channels at once, leading to better overall performance. If the plant has p inputs and q outputs, the objective can be written as

$$\min\left\{\kappa_1\sum_{j=1}^{q}\|\phi_j\|_1 + \kappa_2\sum_{j=1}^{p}\|\bar{u}_j\|_1\right\} \tag{15}$$

For this objective to be suitable for a linear program, the same substitution employed in the SISO case is useful

$$\phi = \phi^+ - \phi^- \qquad \boldsymbol{u} = \boldsymbol{u}^+ - \boldsymbol{u}^- \tag{16}$$

To make the final result more compact, some shorthand will be used. Note that the subscripting for the following M_i terms has been changed from the SISO case to make the subscript ordering more logical:

$$M_1 = \left(hrw_{1_D}\right)I \tag{17}$$

$$M_2 = \left(rw_{1_D}\right)I \tag{18}$$

$$M_3 = -\mathcal{S}(\hat{\boldsymbol{w}}, h\hat{d}) \tag{19}$$

$$M_4 = \mathcal{S}(\hat{\boldsymbol{w}}, \hat{n}) \tag{20}$$

By our using the substitutions of Eqs. (16–20) the MIMO tracking optimization can be expressed as the following linear program:

$$\min\left\{\kappa_1 \sum_{j=1}^{q}\sum_{i=0}^{k-1}\left(\phi_j^+ + \phi_j^-\right)_i + \kappa_2 \sum_{j=1}^{p}\sum_{i=0}^{k-1}\left(\bar{u}_j^+ + \bar{u}_j^-\right)_i\right\}$$

$$\times \begin{bmatrix} \langle M_1\rangle & -\langle M_1\rangle & & & \langle M_3\rangle \\ & & \langle M_2\rangle & -\langle M_2\rangle & \langle M_4\rangle \end{bmatrix} \begin{Bmatrix} \bar{\boldsymbol{u}}^+ \\ \bar{\boldsymbol{u}}^- \\ \phi^+ \\ \phi^- \\ \mathbf{col}\left(N_{C_1}\right) \end{Bmatrix}$$

$$= \begin{Bmatrix} -rw_{1_D}\boldsymbol{u}_{\mathrm{SS}} \\ r\hat{\boldsymbol{w}} \end{Bmatrix} \qquad \bar{\boldsymbol{u}}^+, \bar{\boldsymbol{u}}^-, \bar{\phi}^+, \bar{\phi}^- \geq 0 \tag{21}$$

where the i subscript in the objective refers to the ith coefficient of the terms inside the parentheses.

IV. Regulation

The regulation optimization is concerned with finding N_{C_2} and D_C. Together, these two terms dictate the system's response to disturbances, sensor noise, and modeling errors. In the SISO case, this part of the problem was called the robustness optimization, an extension of the usual meaning that normally only applies to the modeling error component. A more accurate description would be that the controller regulates the response due to these environmental effects, and so to avoid confusion with the conventional use of the term robustness, this optimization has been relabelled the regulation part of the problem.

In the SISO case, converting Eq. (11) into linear programming form was relatively easy because the terms could be reordered as desired. In the MIMO case, however, this is not so because matrices are involved. Some internal reordering of the matrices is necessary to put the constraints in a form suitable for a linear programming optimization. Consider the transfer function matrix relating the disturbances to the actuator demands,

$$u_2 = (dD_C - I)W_2 \tag{22}$$

$$\mathrm{col}_i(u_2) = \mathrm{col}_i(dD_C W_2) - \mathrm{col}_i(W_2) \tag{23}$$

The problem with this is that the unknown D_C is postmultiplied by W_2. Before this constraint can be included in a linear program, all unknowns must only be premultiplied. The diagonal structure of W_2 can be exploited to achieve this

$$\mathrm{col}_i(u_2) = W_{2_i} d\,\mathrm{col}_i(D_C) - \mathrm{col}_i(W_2) \tag{24}$$

$$\begin{Bmatrix} \mathrm{col}_1(u_2) \\ \mathrm{col}_2(u_2) \\ \vdots \end{Bmatrix} = \begin{bmatrix} W_{2_1}d & & \\ & W_{2_2}d & \\ & & \ddots \end{bmatrix} \begin{Bmatrix} \mathrm{col}_1(D_C) \\ \mathrm{col}_2(D_C) \\ \vdots \end{Bmatrix} - \begin{Bmatrix} \mathrm{col}_1(W_2) \\ \mathrm{col}_2(W_2) \\ \vdots \end{Bmatrix} \tag{25}$$

$$\mathbf{col}(u_2) = \mathcal{D}(W_2, d)\,\mathbf{col}(D_C) - \mathbf{col}(W_2) \tag{26}$$

The u_{3-4} and ϕ_{2-4} transfer function matrices can be expressed similarly. Equation (5) suffers a similar problem in that the unknown N_{C_2} and D_C terms are postmultiplied by a matrix, but this time it is not by a diagonal one. The method for converting this into linear programming form is most easily shown by example. Consider two (2×2) matrices, a and b being multiplied together, where the unknown a matrix is postmultiplied by the known matrix b. By the explicit expansion of the operation, the unknowns can be moved to the right

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \equiv \begin{bmatrix} b_{11} & & b_{21} & \\ & b_{11} & & b_{21} \\ b_{12} & & b_{22} & \\ & b_{12} & & b_{22} \end{bmatrix}\begin{Bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{Bmatrix} \tag{27}$$

$$\equiv \mathcal{E}(b^T, 2)\,\mathbf{col}(a) \tag{28}$$

Indeed, the equivalence of Eq. (28) holds for any pair of square matrices, not just (2×2) matrices. When we use this result, for a plant with p inputs, Eq. (5) can be expressed as

$$\mathcal{E}(d^T, p)\,\mathbf{col}(D_C) + \mathcal{E}(n^T, p)\,\mathbf{col}\left(N_{C_2}\right) = \mathbf{col}(I) \tag{29}$$

Two other constraints must also be included in the constraints. The first ensures that the controller stabilizes all plants within the uncertainty set, the condition for which was derived earlier as $\|(I - dD_C)W_4\|_1 \leq 1$. This can be expressed more compactly as

$$\|u_4\|_1 \leq 1 \tag{30}$$

The second and final constraint accounts for the computational delay inherent in any digital controller implementation by setting the z^0 coefficient of all polynomials in N_{C_2} to zero. The reasoning behind this is the same as the SISO case of Ref. 1. In linear programming form, this constraint is equivalent to setting the first term in each N_{C_2} sequence to zero.

The objective of the linear program is formed in the same way as for the SISO case, making the substitutions $u_2 = u_2^+ - u_2^-$ and so on to remove the nonlinear norm operations (the uncertainty constraint is also handled in this way). Unlike conventional l_1 objectives, however, the l_1 norm of each element of the matrices are taken and summed, just as was done for the tracking optimization of the preceding section. For example, for a plant with p inputs and q outputs, ϕ_2 is included in the objective in the following way:

$$\sum_{i=1}^{q}\sum_{j=1}^{p}\left\|\phi_{2_{ij}}\right\|_1 \tag{31}$$

This allows the optimization to minimize the responses in all channels, not just the worst one. Assembling these alterations into a complete linear program leads to Eq. (32).

$$\min \sum_{t=0}^{k-1} \left\{ \begin{array}{l} \kappa_1 \sum_{i=1}^{p}\sum_{j=1}^{p}\left(u_{2_{ij}}^{+} + u_{2_{ij}}^{-}\right)_t + \kappa_2 \sum_{i=1}^{p}\sum_{j=1}^{q}\left(u_{3_{ij}}^{+} + u_{3_{ij}}^{-}\right)_t + \kappa_3 \sum_{i=1}^{p}\sum_{j=1}^{p}\left(u_{4_{ij}}^{+} + u_{4_{ij}}^{-}\right)_t + \kappa_4 \sum_{i=1}^{q}\sum_{j=1}^{p}\left(\phi_{2_{ij}}^{+} + \phi_{2_{ij}}^{-}\right)_t \\ + \kappa_5 \sum_{i=1}^{q}\sum_{j=1}^{q}\left(\phi_{3_{ij}}^{+} + \phi_{3_{ij}}^{-}\right)_t + \kappa_6 \sum_{i=1}^{q}\sum_{j=1}^{p}\left(\phi_{4_{ij}}^{+} + \phi_{4_{ij}}^{-}\right)_t \end{array} \right\}$$

$$\left[\begin{array}{cccccc:cccccc:cc} \langle I\rangle &&&&&& -\langle I\rangle &&&&&& & -\langle\mathcal{D}(W_2,d)\rangle \\ & \langle I\rangle &&&&&& -\langle I\rangle &&&&& \langle\mathcal{D}(W_3,d)\rangle & \\ && \langle I\rangle &&&&&& -\langle I\rangle &&&& & -\langle\mathcal{D}(W_4,d)\rangle \\ &&& \langle I\rangle &&&&&& -\langle I\rangle &&& & \langle\mathcal{D}(W_2,n)\rangle \\ &&&& \langle I\rangle &&&&&& -\langle I\rangle && -\langle\mathcal{D}(W_3,n)\rangle & \\ &&&&& \langle I\rangle &&&&&& -\langle I\rangle & & \langle\mathcal{D}(W_4,n)\rangle \\ &&&&&&&&&&&& \langle\varepsilon(n^T,p)\rangle & \langle\varepsilon(d^T,p)\rangle \\ \hdashline && \xi(p,k) &&&&&& \xi(p,k) &&&& & \end{array}\right] \left\{\begin{array}{c} \mathbf{col}(u_2^+) \\ \mathbf{col}(u_3^+) \\ \mathbf{col}(u_4^+) \\ \mathbf{col}(\phi_2^+) \\ \mathbf{col}(\phi_3^+) \\ \mathbf{col}(\phi_4^+) \\ \mathbf{col}(u_2^-) \\ \mathbf{col}(u_3^-) \\ \mathbf{col}(u_4^-) \\ \mathbf{col}(\phi_2^-) \\ \mathbf{col}(\phi_3^-) \\ \mathbf{col}(\phi_4^-) \\ \mathbf{col}(N_{C_2}) \\ \mathbf{col}(D_C) \end{array}\right\} \begin{array}{c} = \\ \vdots \\ \le \end{array} \left\{\begin{array}{c} -\mathbf{col}(W_2) \\ 0 \\ -\mathbf{col}(W_4) \\ 0 \\ 0 \\ 0 \\ \mathbf{col}(I) \\ \cdots\cdots\cdots \\ 1 \end{array}\right\}$$

$$(N_{C_2})_{z^0} = 0, \qquad u_i^{\pm}, \phi_i^{\pm} \ge 0 \tag{32}$$

When the coprime factorization of the plant is such that $r = l = 1$ and the W_i filters all have denominators of unity, Eq. (32) sufficiently describes the problem. If this is not the case, these terms must be explicitly included when converting to linear programming form. Although the necessary adjustments are relatively simple, they introduce the possibility of conservatism in the results. Consider the uncertainty constraint when r and l are not unity and the W_4 filter has a nonunity denominator, W_{4_D}

$$\|u_4\|_1 \le 1 \tag{33}$$

$$\|(dD_C - I)W_4\|_1 \le 1 \tag{34}$$

$$\left\|\left(1/rlW_{4_D}\right)(\hat{d}\hat{D}_C - rlI)\hat{W}_4\right\|_1 \le 1 \tag{35}$$

The $1/rlW_{4_D}$ term poses significant difficulty when converting to linear program form, so the well-known triangle inequality for gains (see Ref. 6, for example) $\|ab\| \le \|a\|\|b\|$ is used

$$\left\|1/rlW_{4_D}\right\|_1 \|(\hat{d}\hat{D}_C - rlI)\hat{W}_4\|_1 \le 1 \tag{36}$$

All of the terms in the $\|1/rlW_{4_D}\|_1$ norm are known quantities. Using the triangle inequality has the disadvantage that it allows conservatism in the constraint, and so the linear program may constrain the optimization tighter than necessary. Also, a side effect of this is that all of the objective norms also have the potential to be conservative because they, too, must use the triangle inequality. If the degree of conservatism varies significantly between each of the objective norms, then the relative weightings between them, specified by the κ_i, is not reflected in the optimization results.

V. Example

As an example, consider the large space structure (LSS) studied in Refs. 7 and 8. This plant is modeled on a structure with 18 actuators and 20 sensors, which, through strategic groupings and placement has a two-input/two-output linear model. The inputs are the actuator groups at the primary and secondary mirrors and the outputs are the line-of-sight sensors, also at the mirrors. For the purpose of illustration, the maneuver chosen was a 0.05-rad step change in the primary mirror angle, and a 0.03-rad step change at the secondary mirror. This was to be achieved with optimal actuator demands not exceeding 1.0 in the first input channel and 0.8 in the second. The emphasis of the design was to be on the tracking accuracy, but the actuator demands were not to be excessive.

Because the LSS model has no poles on the stability boundary, a step change in the plant output can only be maintained if a steady-state actuator input is applied. This was obtained using the discrete state-space matrices of the plant (discretized at 1000 Hz with a zero-order hold)

$$\left\{\begin{array}{c} \boldsymbol{x}_{ss} \\ \boldsymbol{y}_{ss} \end{array}\right\} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \left\{\begin{array}{c} \boldsymbol{x}_{ss} \\ \boldsymbol{u}_{ss} \end{array}\right\} \tag{37}$$

$$\left\{\begin{array}{c} \boldsymbol{x}_{ss} \\ \boldsymbol{u}_{ss} \end{array}\right\} = \begin{bmatrix} (A - I) & B \\ C & D \end{bmatrix}^{-1} \left\{\begin{array}{c} 0 \\ \boldsymbol{y}_{ss} \end{array}\right\} \tag{38}$$

$$\boldsymbol{u}_{ss} = \left\{\begin{array}{c} 0.0502 \\ 0.0301 \end{array}\right\} \tag{39}$$

Using the continuous state-space matrices instead leads to a similar matrix equation (the $-I$ disappears) with the same result. The only other information needed for the tracking optimization was the objective weightings κ_1 and κ_2. The tracking accuracy was considered more important, and so $\kappa_1 = 20$ and $\kappa_2 = 1$ were chosen to reflect this. Performing the optimization with the ϕ and $\bar{\boldsymbol{u}}$ sequences restricted to 100 terms gave the optimal results of Fig. 2.

The regulation optimization requires the operating environment of the plant to be characterized. It was assumed that the environment of the LSS contained disturbances and a low level of sensor noise, both of which had known maximums but unknown frequency contents. They were, therefore, represented by constant W_2 and W_3 matrices. Assuming the disturbances were no greater than 0.1 in each input channel and that the sensor noise was no worse than 0.002, the weighting matrices were expressed as

$$W_2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{40}$$

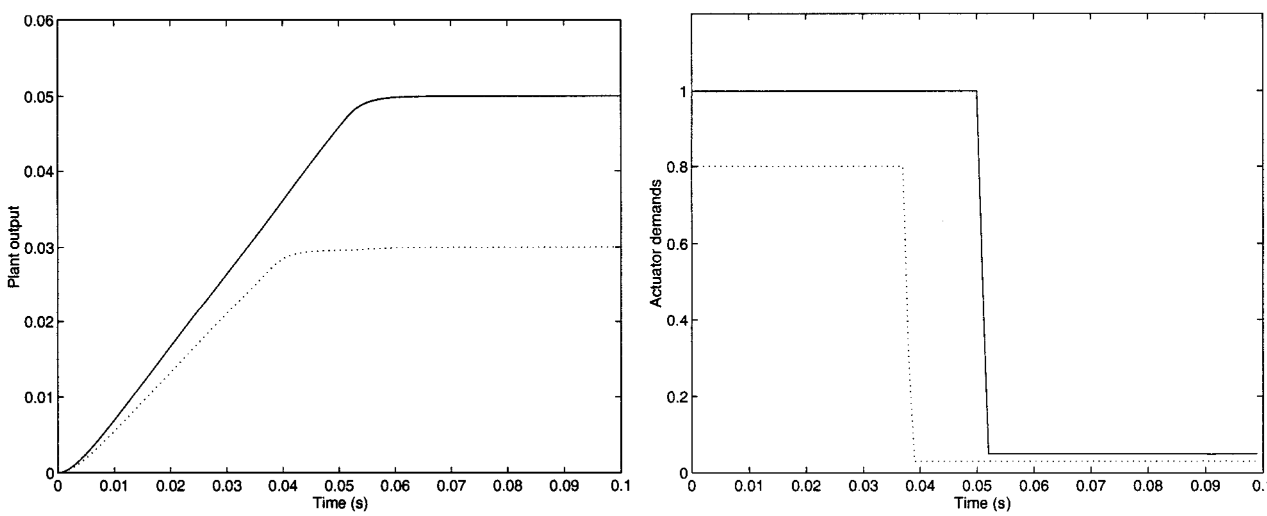
$$W_3 = \begin{bmatrix} 0.002 & 0 \\ 0 & 0.002 \end{bmatrix} \tag{41}$$

Table 1 Results of high-order regulation optimization, ϕ emphasized

Norm	w_{2_1}	w_{2_2}	w_{3_1}	w_{3_2}	w_{4_1}	w_{4_2}
$\|u_{1j}\|_1$	9.9213e−002	2.2938e−003	1.2073e+000	6.2458e−002	5.5229e−001	1.8609e−002
$\|u_{2j}\|_1$	3.0784e−003	9.8004e−002	9.9599e−002	8.2745e−001	1.9638e−002	4.3292e−001
$\|\phi_{1j}\|_1$	7.8445e−004	1.6366e−005	1.9843e−003	4.5598e−005	3.3946e−003	1.2500e−004
$\|\phi_{2j}\|_1$	2.3028e−005	2.0017e−003	5.8119e−005	1.9601e−003	1.9438e−004	2.6655e−003

Table 2 Results of low-order regulation optimization, ϕ emphasized

Norm	w_{2_1}	w_{2_2}	w_{3_1}	w_{3_2}	w_{4_1}	w_{4_2}
$\|u_{1j}\|_1$	9.8540e−002	1.0292e−003	2.9745e−001	1.6155e−002	3.3758e−001	4.4295e−003
$\|u_{2j}\|_1$	1.1517e−003	9.8549e−002	2.2786e−002	2.9437e−001	5.2493e−003	2.8634e−001
$\|\phi_{1j}\|_1$	1.4556e−003	4.9506e−005	1.9708e−003	2.0350e−005	4.6562e−003	2.0542e−004
$\|\phi_{2j}\|_1$	6.8490e−005	1.4472e−003	2.3190e−005	1.9710e−003	3.3217e−004	3.9526e−003



Plant outputs **Actuator demands**

Fig. 2 Optimal results for the LSS tracking optimization.

The plant uncertainty was expected to be significant, given that the model was a reduction from a considerably larger one. For the sake of discussion, different uncertainty profiles were used on the two input channels

$$W_{4_1}(s) = (s + 10)/(s + 400) \qquad W_{4_2}(s) = (s + 20)/(s + 500) \tag{42}$$

These specify similar higher-frequency uncertainty, but the low-frequency characteristics are different. The W_4 weighting matrix, therefore, had the following representation, which was then discretized

$$W_4(s) = \frac{1}{(s + 400)(s + 500)} \times \begin{bmatrix} (s + 10)(s + 500) & 0 \\ 0 & (s + 20)(s + 400) \end{bmatrix} \tag{43}$$

$\|w_4\|_\infty$ was needed before the objective weightings could be chosen to make κ_3 and κ_6 the correct magnitudes relative to the other κ_i. This was found in the same way as for the SISO case

$$\|w_4\|_\infty = \|W_2\|_\infty + \|u\|_\infty \tag{44}$$

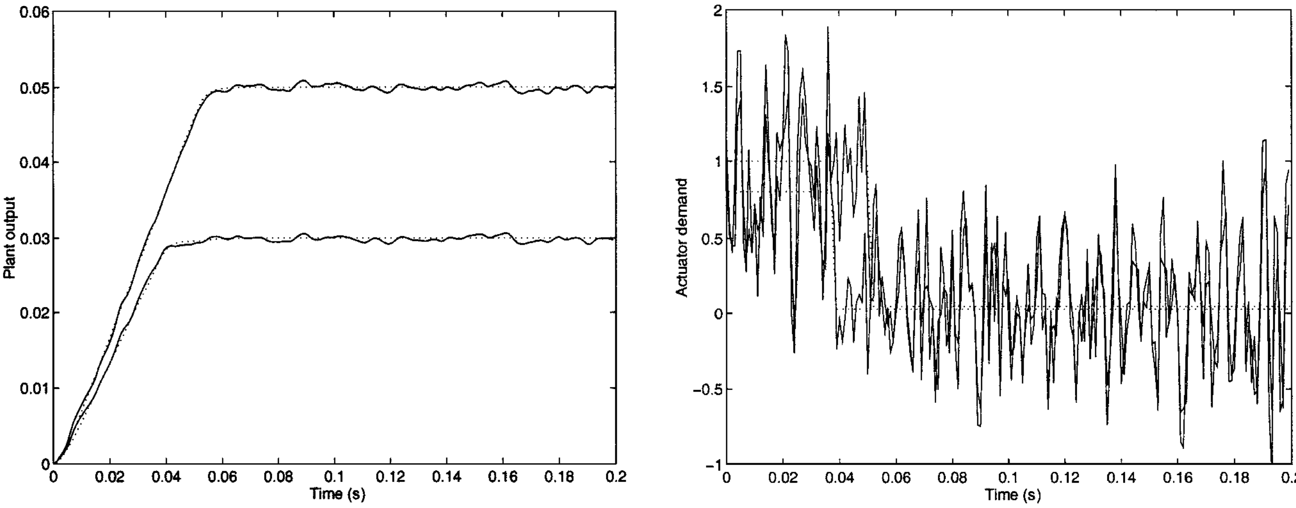
where the ∞ subscript refers to the l_∞ norm. $\|W_2\|_\infty$ is 0.1 in both input channels, and the maximum optimal actuator demand is 1 in the first channel and 0.8 in the second. Allowing actuator demands up to roughly 1.4 and 1.6, respectively, for regulating the disturbances, etc., then $\|W_4\|_\infty$ is approximately 2.5 in both input channels. We emphasized the tracking error more heavily than the actuator demands (this time 1000 times greater) using the following objective weightings

$$\kappa_1 = 1 \qquad \kappa_2 = 1 \qquad \kappa_3 = 2.5$$

$$\kappa_4 = 1{,}000 \qquad \kappa_5 = 1{,}000 \qquad \kappa_6 = 2{,}500 \tag{45}$$

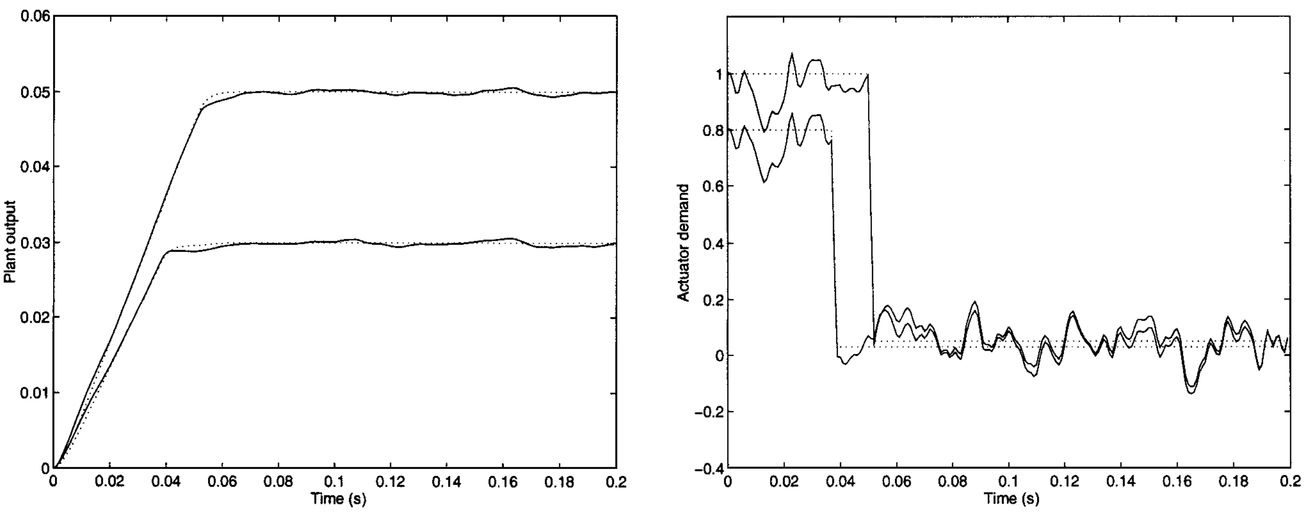
Two regulation optimizations were performed. The first was solved with N_{C_2} and D_C restricted to 150 terms in their polynomial entries and $r = l = 1$. The W_i filters were also replaced with their impulse responses, giving $W_{i_D} = 1$. This removed all conservatism in the results. The second optimization used r and l obtained from the coprime factorization algorithm outlined in Refs. 2, 3, and 9, and the $\hat{N}_{C_2}$ and $\hat{D}_C$ polynomial entries were restricted to five terms. The degree of conservatism exhibited by this low-order solution was very high, with the conservative objective norms ranging from 100 up to 10,000 times greater than the true values (as found by taking l_1 norms of the impulse responses of the norm transfer functions). The uncertainty constraint was relaxed because it was observed that the true u_4 norm was less than 1, even though the equivalent conservative norm was not. Despite the high degree of conservatism, the true norms were comparable with the high-order, exact optimization results. The norms for these optimizations can be seen in Tables 1 and 2.

Simulations were performed with both the high- and low-order controllers using a perturbed plant model. To ensure that the perturbed model was within the uncertainty set ($\|\Delta\|_1 < 1$), Δ was taken to be 0.5. The disturbances and sensor noise were generated by passing white noise through the W_2 and W_3 filters, respectively. The results of Figs. 3 and 4 show that good tracking accuracy was maintained despite the high level of disturbances and the inaccurate plant model, but the actuator demands of the high-order controller were far more excessive than those of the low-order controller.



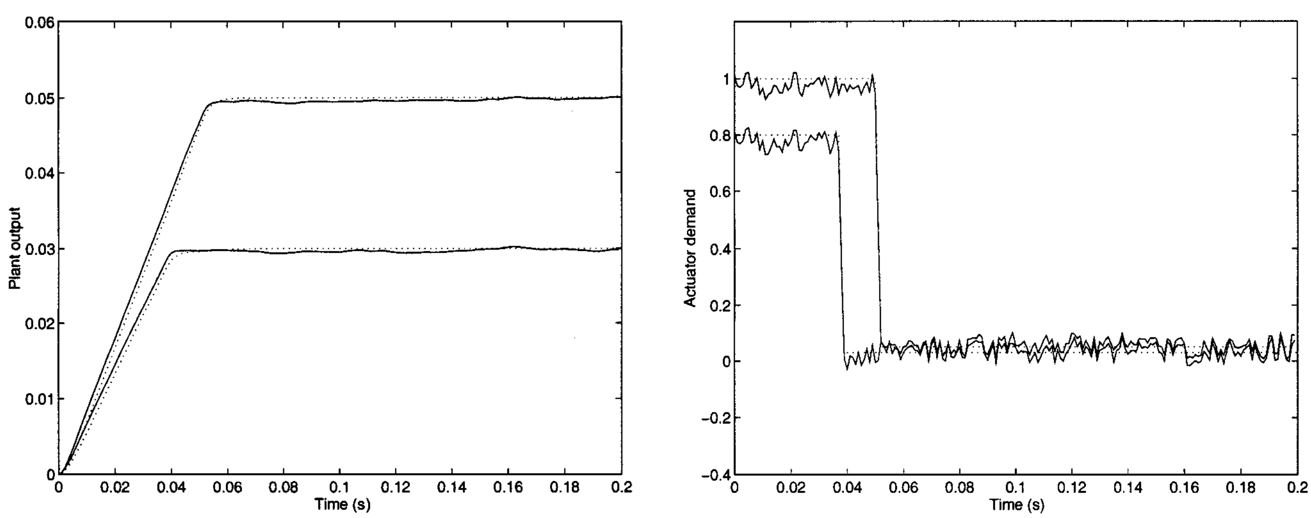
Plant outputs Actuator demands

Fig. 3 Simulation with high-order controller, ϕ emphasized.



Plant outputs Actuator demands

Fig. 4 Simulation with low-order controller, ϕ emphasized.



Plant outputs Actuator demands

Fig. 5 Simulation with high-order controller, less ϕ emphasis.

Table 3 Results of high-order regulation optimization, less ϕ emphasis

Norm	w_{2_1}	w_{2_2}	w_{3_1}	w_{3_2}	w_{4_1}	w_{4_2}
$\|u_{1j}\|_1$	9.5738e−002	2.3965e−004	7.3546e−002	4.7308e−003	6.5756e−002	6.1036e−004
$\|u_{2j}\|_1$	5.0610e−004	9.5712e−002	6.6214e−003	7.7715e−002	1.4650e−003	5.2663e−002
$\|\phi_{1j}\|_1$	4.2534e−003	3.5877e−005	1.9148e−003	8.0648e−006	4.5811e−003	1.4003e−004
$\|\phi_{2j}\|_1$	4.5752e−005	4.4032e−003	1.3683e−005	1.9142e−003	2.2415e−004	3.3283e−003

Therefore, a second set of regulation optimizations with a lower emphasis on ϕ was performed, this time with $\kappa_4 = 100$, $\kappa_5 = 100$, and $\kappa_6 = 250$. The low-order results were exactly the same as before, but the high-order controller showed much improved actuator activity and slightly better tracking accuracy (see Fig. 5). This was expected because the actuator norms, in particular, were much lower (see Table 3).

Note that, as was the case in the SISO example of Ref. 1, the regulation optimizations were solved by converting them to their dual linear programs first. This was necessary because the problems as formulated proved numerically difficult to solve. Unlike the SISO example, however, the tracking optimization did not require this step, although the feasibility tolerance had to be relaxed a little to reach a solution. The dual tracking problem did not require this to be done.

VI. Conclusions

The method of Ref. 1 has been successfully extended to plants with any number of inputs and outputs. Conditions ensuring that the controller stabilizes the system have been derived, including a proof based on the small gain theorem for multiplicative plant uncertainty with the two-degree-of-freedom controller arrangement. It has been shown by the numerical example that the method is effective in generating controllers for real-world environments, even where the plant model is not exact. The results also show that both low- and high-order controllers can be successfully formed and used to obtain good system performance.

Appendix: Operator Notation

For $\mathrm{col}_i(a)$, extract the ith column of a

$$\mathbf{col}(a) = \begin{bmatrix} \mathrm{col}_1(a) \\ \mathrm{col}_2(a) \\ \vdots \end{bmatrix}$$

For $\mathcal{D}(a, b)$, a is a diagonal matrix and b can be any matrix. The operator multiplies the entire b matrix by each diagonal element of a in turn, storing the results diagonally. The definition is best shown by example:

$$\begin{bmatrix} a_{11}b & & \\ & a_{22}b & \\ & & \ddots \end{bmatrix}$$

For $\mathcal{E}(a, i)$, expand each element of the matrix a into an $(i \times i)$ diagonal submatrix. The result will have i times as many rows and columns as the original a matrix. For example,

$$\mathcal{E}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right) = \begin{bmatrix} 1 & & 2 & \\ & 1 & & 2 \\ 3 & & 4 & \\ & 3 & & 4 \end{bmatrix}$$

$\mathcal{S}(\boldsymbol{a}, b)$ is similar to $\mathcal{D}(\boldsymbol{a}, b)$ except that $\boldsymbol{a}$ is a vector and the results are stored side by side instead of diagonally. For example, if there are q elements in the $\boldsymbol{a}$ vector, the result becomes $[a_1 b, a_2 b, \ldots, a_q b]$

For $\xi(p, k)$, there are p blocks (side by side) of p rows, each consisting of k 1's. For example,

$$\xi(2, 3) = \begin{bmatrix} 1 & 1 & 1 & & & & 1 & 1 & 1 & & & \\ & & & 1 & 1 & 1 & & & & 1 & 1 & 1 \end{bmatrix}$$

For $\langle x \rangle$, convert polynomial matrix x to its numerical convolution equivalent. For example, if a–f are second-order polynomials and the matrix convolution

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{Bmatrix} e \\ f \end{Bmatrix}$$

is to be converted into a numerical operation, then

$$\left\langle \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right\rangle = \begin{bmatrix} a_0 & & & b_0 & & \\ a_1 & a_0 & & b_1 & b_0 & \\ a_2 & a_1 & a_0 & b_2 & b_1 & b_0 \\ & a_2 & a_1 & & b_2 & b_1 \\ & & a_2 & & & b_2 \\ c_0 & & & d_0 & & \\ c_1 & c_0 & & d_1 & d_0 & \\ c_2 & c_1 & c_0 & d_2 & d_1 & d_0 \\ & c_2 & c_1 & & d_2 & d_1 \\ & & c_2 & & & d_2 \end{bmatrix}$$

Acknowledgment

This work has been associated with a research Grant from the Australian Research Council.

References

[1]Scott, C. N., and Wood, L. A., "Optimal Robust Tracking Subject to Disturbances, Noise, Plant Uncertainty, and Performance Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 774–779.

[2]Vidyasagar, M., *Control System Synthesis: A Factorization Approach*, MIT Press, Cambridge, MA, 1985, Chaps. 4 and 5.

[3]Vethecan, J., "Design of Compensators for LTI Feedback Systems by Convex Optimization," Ph.D. Thesis, Dept. of Aerospace Engineering, Royal Melbourne Inst. of Technology, Melbourne, VIC, Australia, 1996, Chaps. 3–6.

[4]Dahleh, M. A., and Pearson, J. B., Jr., "Optimal Rejection of Persistent Disturbances, Robust Stability, and Mixed Sensitivity Minimization," *IEEE Transactions on Automatic Control*, Vol. 33, No. 8, 1988, pp. 722–731.

[5]Dahleh, M. A., and Ohta, Y., "A Necessary and Sufficient Condition for Robust BIBO Stability," *Systems & Control Letters*, Vol. 11, 1988, pp. 271–275.

[6]Boyd, S. P., and Barratt, C. H., *Linear Controller Design, Limits of Performance*, Prentice–Hall, Englewood Cliffs, NJ, 1991, Chaps. 5 and 10.

[7]Chiang, R. Y., and Safonov, M. G., *Robust Control Toolbox Users Guide*, Math Works, Natick, MA, 1992, pp. 64–69.

[8]Safonov, M. G., Chiang, R. Y., and Flashner, H., "H^∞ Control Synthesis for a Large Space Structure," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 3, 1991, pp. 513–520.

[9]Scott, C. N., "l_1-Optimal Robust Tracking," Ph.D. Thesis, Royal Melbourne Inst. of Technology, Melbourne, VIC, Australia, Appendix A, (submitted).