2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Model Synthesis Weighting Effects on Model Tuning in System Identification

Sonny Nimityongskul[1], Seth Lacy[2] and Vit Babuska[3]

Abstract— **System identification is the process of deriving dynamic equations from observed system behavior, the inverse of the common problem of deriving solutions to a given set of dynamics. The system identification process generally consists of two steps, a model synthesis step followed by a model tuning step. For complex systems, standard system identification tools often fail to provide satisfactory results without extensive manipulation by an experienced engineer. Input, output, and frequency weightings are often used to adjust the properties of the identified model in model tuning. In this effort, we examine the impact of model synthesis weightings on model tuning results. Model synthesis weightings are shown to improve the initial models used for model tuning. However, it is shown that an improved initial model for model tuning does not necessarily lead to faster model tuning or more accurate identified models.**

I. Introduction

System identification is the process by which models are derived from measured data. System identification is often used to derive models for controller design. The system identification process generally consists of two steps: a model synthesis step followed by a model tuning step. In the model synthesis step an initial model is constructed directly from the measured data. In the tuning step the synthesized model is updated based on a user specified cost function in an optimization algorithm. In general, model synthesis is computationally faster than model tuning.

Standard system identification tools often fail to provide satisfactory results for complex systems without extensive modification by an experienced engineer [1,3,6,9]. In particular, input, output, and frequency weightings are used to affect the properties of the resulting model. These weightings are often applied only in the model tuning phase of system identification.

In this effort, various weighting techniques are applied to both model synthesis and model tuning to create high-fidelity state space models for use in controller design. To facilitate this comparison, analytic gradients are computed for a general least squares model tuning cost function. It is the goal of this research to show that model synthesis weighting can improve the initial model used in model tuning. An improved initial model can result in a significant reduction in model tuning time, in turn reducing the overall time of the entire system identification process.

This work was supported by the Air Force Office of Scientific Research under LRI00VS17COR.

[1]University of Wisconsin-Madison, apnimityongs@wisc.edu
[2]Air Force Research Laboratory, seth.lacy@kirtland.af.mil
[3]Sandia National Laboratory, vbabusk@sandia.gov

II. Model Synthesis

A. FORSE

A number of subspace based techniques have been developed to create state space models from measured data [1-4]. The FORSE identification algorithm was chosen for this study because it has been shown to be effective at producing accurate high dimensional models. The FORSE identification algorithm is based on a singular value decomposition of the extended observability matrix derived from frequency domain data [5]. FORSE has a built in frequency weighting scheme, and can use non-uniformly spaced frequency data.

B. Weightings for Model Synthesis

Weightings are implemented when standard system identification tools fail to provide a model that meets desired accuracy or other characteristics. The goal of weighting during model synthesis is to improve the initial model used in model tuning. Two types of weighting schemes were used in model synthesis, input-output weighting and frequency weighting.

The frequency weighting in the FORSE algorithm allows the user to apply a specific weighting at each input for each frequency point [5]. Due to the construction of the weighting scheme the input channels can be weighted independently, but all of the outputs carry the same weight.

Modelers can weight the individual frequencies in any manner they deem fit. The two weighting techniques used in this study were to apply a constant weight over a frequency band, and weighting by the inverse of the standard deviation of the measured data. A constant frequency band weighting can be used to reduce or amplify model errors in a specific frequency band. This technique may be used to improve model accuracy near zeros, which are often important for control design. This can provide an improved model fit in the weighted frequency band, often at the expense of larger model errors elsewhere. Weighting by the inverse of the variance or standard deviation ensures that the data points with the lowest uncertainty receive the highest weight. Since only the input channels can be weighted in FORSE, the variance weighting was based on the average of the variance data over all of the output channels.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

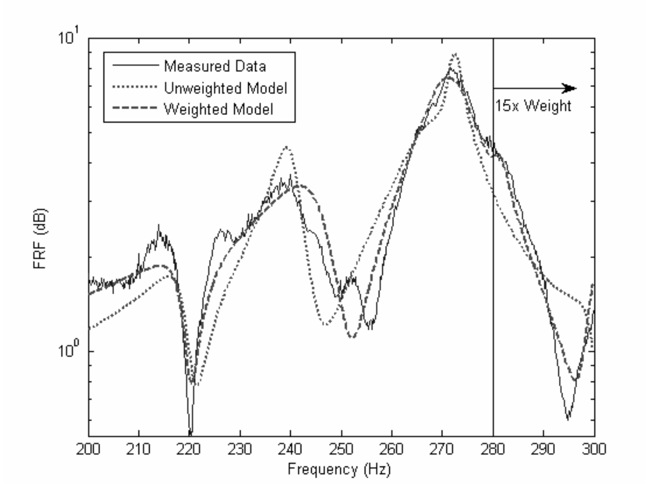

Figure 1: FRF for unweighted (dotted) and weighted (dashed) models

The goal of input-output weighting is to improve the identified model by adjusting the relative signal levels between various input-output channels [6]. Input-output weighting is only applicable to MIMO systems. For system identifications algorithms which are sensitive to the scale of the data, such as FORSE, proper input-output weighting can result in an improved identified model.

In the frequency domain the input-output weighting matrices are applied at each frequency point,

$$\hat{G}_w(\omega) = W_{out}\hat{G}(\omega)W_{in} \quad (1)$$

where $W_{in} \in \Re^{m\times m}$ and $W_{out} \in \Re^{p\times p}$ are the weighting matrices for the input and output channels, respectively, and $\hat{G}(\omega), \hat{G}_w(\omega) \in C^{p\times m}$ are the unweighted and weighted frequency response functions. There are numerous techniques for selecting W_{out} and W_{in} [6]. The weighted frequency response can be used in model synthesis, model tuning, or both the synthesis and tuning steps. The final weighted model must be unweighted to obtain the identified model in natural (original) coordinates. For a state space model, this is done by

$$\begin{aligned} A &= A_w \\ B &= B_w W_{in}^{-1} \\ C &= W_{out}^{-1} C_w \\ D &= W_{out}^{-1} D_w W_{in}^{-1} \end{aligned} \quad (2)$$

where $A, A_w \in \Re^{n\times n}$, $B, B_w \in \Re^{n\times m}$, $C, C_w \in \Re^{p\times n}$, and $D, D_w \in \Re^{p\times m}$ represent the natural and weighted system matrices. Input-output weighting must be used judiciously because it essentially warps the space from which model parameters are extracted. The net effect of input-output weighting on the final unscaled identified model may be smaller errors in the low response paths at the expense of larger errors in the high response paths.

III. Model Tuning

Model synthesis may produce a suboptimal estimate of the state space system. To improve the initial model, it can be tuned to a specified metric [7].

A. Cost Functions

The accuracy of an identified model can be gauged by the difference between the measured and modeled frequency response functions

$$J = \sum_{k=1}^{K} \left\| \hat{G}(\omega_k) - (C(e^{j\omega_k \Delta t} I - A)^{-1} B + D) \right\|_F^2 \quad (3)$$

where $\Delta t \in \Re$ is the sampling time interval, and $\|\cdots\|_F$ represents the Frobenius norm. The least squares optimization minimizes the absolute error between measured and modeled FRF, so (3) naturally places a greater emphasis on the high magnitude response while devaluing the zeros. An input-output weighting scheme can be used to alter the resulting model by varying the cost associated with each input-output pair. Alternatively, there are a number of other cost functions that can be used to even out the scaling of the data, such as correlation-based metrics or the logarithmic least squares error metric

$$J_{\log} = \sum_{k=1}^{K} \left\| \log(\hat{G}(\omega_k)) - \log(C(e^{j\omega_k \Delta t} I - A)^{-1} B + D) \right\|_F^2 . \quad (4)$$

An optimization scheme was by developed in [1] that used the log least squares cost function in the early tuning iterations to better fit the low magnitude response associated with the transfer function zeros. The scheme then uses the cost function (3) to tune the final model.

B. Correlation Metrics

Correlation metrics are used to compare two vectors. In this work the Frequency Response Assurance Criterion (FRAC) [7] is the correlation metric. The FRAC is a SISO correlation metric with a scalar value between 0 and 1, with perfect correlation indicated by a FRAC value of 1. The FRAC can be generalized into a MIMO correlation metric in several ways, including by creating a matrix of FRAC values for each input-output pair [7]. For the examples used in this work the FRAC values were averaged over all input-output pairs [7]

$$FRAC = \frac{1}{mp}\sum_{i=1}^{m}\sum_{j=1}^{p} \frac{\left|\hat{G}_{j,i}^H G_{j,i}\right|^2}{\left(G_{j,i}^H G_{j,i}\right)\left(\hat{G}_{j,i}^H \hat{G}_{j,i}\right)} . \quad (5)$$

The FRAC metric effectively neutralizes the differences in the scaling of the data, and places importance on both the poles and zeros of the frequency response functions.

C. Parameterization

Selecting the proper model parameterization can lead to tremendous time savings in the model tuning step. Using a straight forward full parameterization is the simplest technique, but is greatly over parameterized and time is wasted tuning a large number of parameters. A minimal parameterization can be created from the real-modal state space form [3,8], in which A takes a form similar to

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \alpha & \beta & 0 \\ 0 & -\beta & \alpha & 0 \\ 0 & 0 & 0 & \lambda_2 \end{bmatrix}, \quad (6)$$

where λ_1, λ_2, $\alpha+i\beta$, and $\alpha-i\beta$ .are the eigenvalues of the state matrix. Modal canonical form places the real eigenvalues of the A matrix on the diagonal and the complex conjugate eigenvalues in real 2x2 blocks along the diagonal of A. Note that the entries of A in modal canonical form (6) are dependent. This dependence in A creates difficulties later on when calculating the derivative of the A matrix for use in the optimization.

A tridiagonal modal parameterization can be used to avoid the dependence of the complex conjugate pairs in the A matrix [9]. The tridiagonal parameterization treats each element on the tridiagonal as a free parameter during tuning. This technique over parameterizes the system by 2n-2 parameters. One advantage of this parameterization is that it allows the tuning algorithm to determine how many of the final eigenvalues are real and complex conjugate pairs.

D. Optimization

The Matlab function *fminunc* from the Optimization Toolbox was used to optimize the models in the model tuning step. The *fminunc* function finds a minimum of a scalar function of several unconstrained variables, starting with an initial estimate obtained in the model synthesis step. If analytic derivatives are not supplied, the optimization can entail a large number of function evaluations and therefore be very time consuming.

To improve the model tuning speed the analytical gradients of the least squares cost function (3)

$$\begin{aligned} J &= \sum_{k=1}^{K} (\hat{G}(\omega_k) - G(\omega_k))^2 \\ &= \sum_{k=1}^{K} (\hat{G}(\omega_k) - G(\omega_k))(\hat{G}(\omega_k) - G(\omega_k))^T \end{aligned} \quad (7)$$

were evaluated. Equation (7) can be expanded to

$$\begin{aligned} J = \sum_{k=1}^{K} \Big(& \hat{G}(\omega_k)\hat{G}(\omega_k)^T - G(\omega)\hat{G}(\omega_k)^T \\ & - \hat{G}(\omega_k)G(\omega_k)^T + G(\omega_k)G(\omega_k)^T \Big). \end{aligned} \quad (8)$$

Writing the model frequency response as

$$G(\omega_k) = C(e^{i\omega_k \Delta t} I - A)^{-1} B + D \quad (9)$$

we can compute the partial derivatives of J with respect to A, B, C, and D as

$$\begin{aligned} \frac{\partial J}{\partial A} &= \sum_{k=1}^{K} 2\Big((e^{i\omega_k \Delta t} I - A)^{-1} B B^T (e^{i\omega_k \Delta t} I - A)^{-T} C^T C (e^{i\omega_k \Delta t} I - A)^{-1} \\ &\qquad + (e^{i\omega_k \Delta t} I - A)^{-1} B (D^T - G(\omega_k)^T) C (e^{i\omega_k \Delta t} I - A)^{-1}\Big)^T \\ \frac{\partial J}{\partial B} &= \sum_{k=1}^{K} 2\Big((D^T - G(\omega_k)^T) C (e^{i\omega_k \Delta t} I - A)^{-1} \\ &\qquad + B^T (e^{i\omega_k \Delta t} I - A)^{-T} C^T C (e^{i\omega_k \Delta t} I - A)^{-1}\Big)^T \\ \frac{\partial J}{\partial C} &= \sum_{k=1}^{K} 2\Big((e^{i\omega_k \Delta t} I - A)^{-1} B (D^T - G(\omega_k)^T) \\ &\qquad + (e^{i\omega_k \Delta t} I - A)^{-1} B B^T (e^{i\omega_k \Delta t} I - A)^{-T} C^T\Big)^T \\ \frac{\partial J}{\partial D} &= \sum_{k=1}^{K} 2\Big(D - G(\omega_k) + C(e^{i\omega_k \Delta t} I - A)^{-1} B\Big), \end{aligned} \quad (10)$$

where $\frac{\partial J}{\partial A} \in \Re^{n\times n}$, $\frac{\partial J}{\partial B} \in \Re^{m\times n}$, $\frac{\partial J}{\partial C} \in \Re^{n\times p}$, and $\frac{\partial J}{\partial D} \in \Re^{p\times m}$. The above gradients were used with the tridiagonal parameterization. The derivation of gradients for other cost functions, such as logarithmic least squares (4) as well as the derivation of Hessians are left for future work.

IV. Results: Deployable Optical Telescope (DOT)

The Deployable Optical Telescope (DOT), as shown in Figure 2, is a space traceable sparse-aperture telescope developed by the Air Force Research Laboratory. DOT is being used to develop and evaluate critical technologies for use in future large space telescopes. DOT has 10 actuators and 9 sensors for high-bandwidth control. The three primary mirrors each have three actuators and a tenth actuator is located at the base of the secondary mirror support tower. There are three sensors for each primary mirror measuring piston, tip, and tilt [10].

Figure 2: Deployable Optical Telescope

The frequency response data used in this work contains 10 inputs and 9 outputs for 3011 frequency points ranging from 5 to 400 Hz. To determine if proper weighting schemes could reduce model tuning time, smaller pieces of the FRF data were used in the system identification. Numerous trials were performed using abbreviated frequency bands of up to 100 Hz with two or three inputs and outputs. For each trial, a model was identified for the unweighted system. Next, a weighting scheme, which used frequency weighting, input-output weighting, or a combination of both, was applied to the FRF data and a second model was identified. If the initial cost (3) of the weighted model was less than the cost of the unweighted model, both models were tuned to determine if the synthesis weighting produced more accurate final models or reduce tuning times.

The results from four trials are discussed below to illustrate the variety of results that were obtained from the model weightings. The reader should only compare the values for the costs functions and tuning times on a trial by trial basis, since the cost function magnitudes vary greatly by input-output pair and the tuning times are highly dependent on the number of states used in the model.

A. *Trial 1: MIMO with Input-Output Weighting*

Trial 1 was a MIMO system using two inputs and three outputs for a frequency range of 10-100 Hz. The model was synthesized with 13 states, which is a low order estimate allowing for greatly reduced tuning time. Input-output weighting was used in Trial 1 to increase the response of the second input channel and the first and third output channels. The weighting reduced the initial cost (3) by approximately 10%. Trials were carried out for three test cases: no weighting, weightings in just the synthesis step, and weighting in both the synthesis and tuning steps. The FRFs for one of the input-output pairs are shown in Figure 3. In Figure 3 the tuned models for the unweighted and synthesis only weighting are shown as one model, because both models converged to the same solution after tuning.

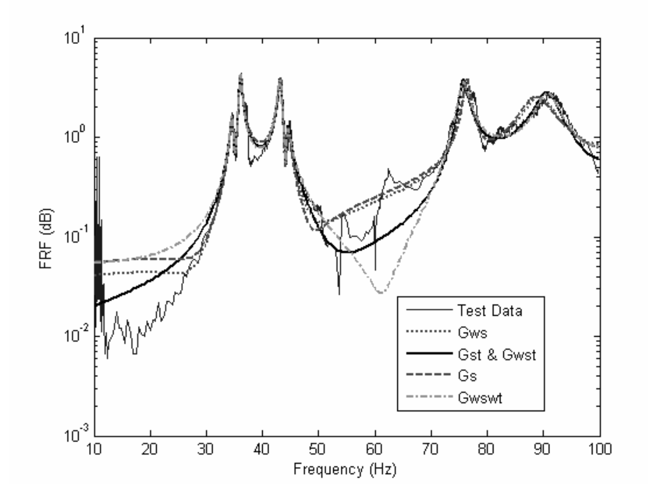

Figure 3: FRF for Trial 1

The results from Trial 1 are summarized in Table 1. For clarity, we adopt the following naming convention: G_s is the unweighted synthesized model, G_{ws} is the weighted synthesized model, G_{st} is the tuned unweighted model, G_{wst} is the tuned weighted model, and G_{wswt} is the weighted model with weighted tuning.

Trial 1	J	FRAC	Tuning Time (s)
G_s	1803	.653	--
G_{ws}	1616	.650	--
G_{st}	1166	.687	802
G_{wst}	1166	.687	589
G_{wswt}	1225	.711	714

Table 1: Costs and tuning times for Trial 1

In Trial 1, both of the models that were unweighted in the tuning step converged to the same solution, with the weighted model having a time savings of 27%. For this Trial, the improvement in initial cost was accompanied by an improvement in tuning time. The model that was weighted in both steps had the highest post tuning cost, but it also had the highest FRAC value. The use of weighting in the tuning step improved the correlation of the low magnitude FRF's by increasing their value in the cost function.

B. *Trial 2: SIMO with Frequency and Output Weighting*

Trial 2 was a SIMO case with frequency weighting and output weighting. Trial 2 used 15 states to identify a model from 100-200Hz. Figure 4 shows the FRF's for one of the two input-output pairs. In Figure 4 only one tuned model is shown because the unweighted and weighted models converged to the same solution after tuning.

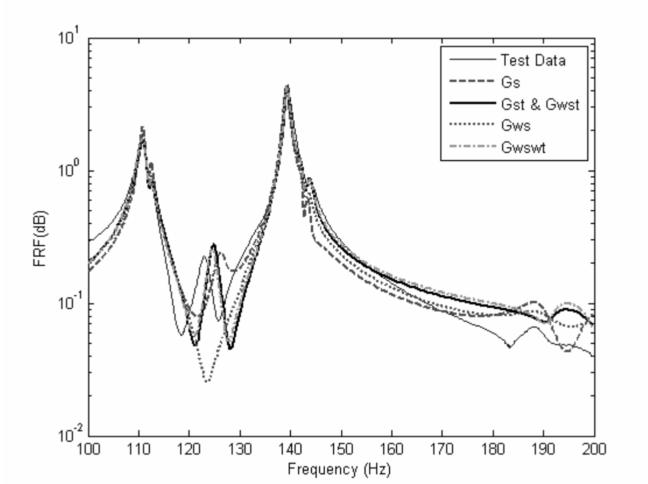

Figure 4: FRF for Trial 2

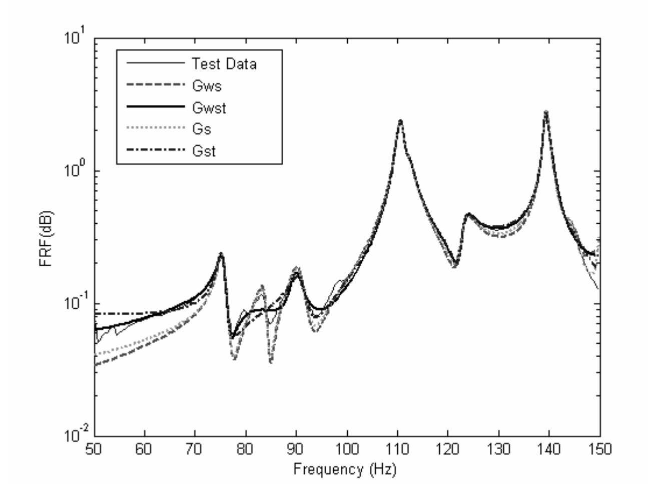

Figure 5: FRF for Trial 3

Table 2 shows that after tuning the unweighted and weighted synthesis models converge to the same solution. Unlike in Trial 1, even though the weighted model had a lower initial cost it took more time for it to converge to the same final solution as the unweighted model. This shows that a lower initial cost does not directly lead to time savings in the tuning step. Additionally, Table 2 shows that the model with weighting in the synthesis and tuning steps had the best FRAC value, but also the highest final cost.

Trial 2	J	FRAC	Tuning Time (s)
G_s	627.9	0.955	--
G_{ws}	603.4	0.981	--
G_{st}	288.3	0.986	668
G_{wst}	288.3	0.986	696
G_{wswt}	297.6	0.990	670

Table 2: Costs and tuning times for Trial 2

C. Trial 3: SISO with Frequency Weighting

Trial 3 used 20 states to identify a SISO model from 50-150Hz for input 3 and output 3. Trial 3 used a frequency weighting to reduce the initial cost by 17%. However, after tuning the unweighted model had a slightly lower cost as compared to the tuned weighted model. The lower final cost came at the expense of a much increase tuning time, which took roughly 60% longer for the unweighted model. The results from this trial show that a reduction in the initial cost does not guarantee a lower cost tuned model.

Trial 3	J	FRAC	Tuning Time (s)
G_s	4.716	.994	--
G_{ws}	3.897	.995	--
G_{st}	0.850	.999	7586
G_{wst}	0.856	.999	4799

Table 3: Costs and tuning times for Trial 3

D. Trial 4: SISO with Frequency Weighting

Trial 4 used 10 states to identify a SISO model from 200-300Hz for input 1 and output 1. The frequency weighting in this trial was able to reduce the initial cost by 75%. However, the weighted model had a greater final cost after the model tuning step. In this case, the tuning of the weighted model reached a local minimum that had a higher cost as compared to the tuned unweighted model.

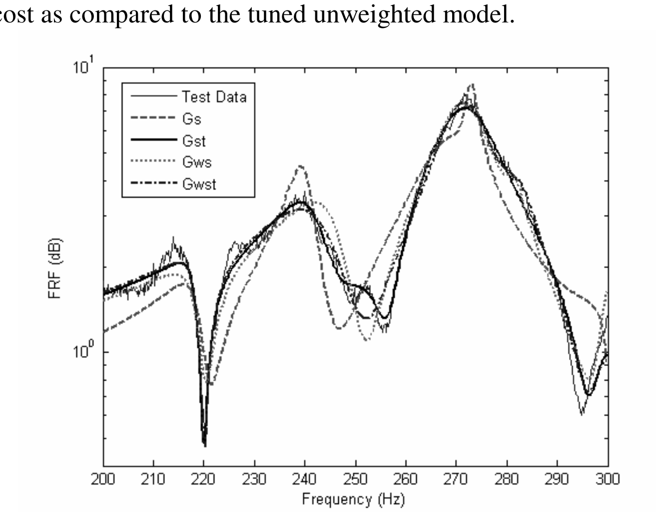

Figure 6: FRF for Trial 4

Trial 4	J	FRAC	Tuning Time (s)
G_s	362.1	.927	--
G_{ws}	91.7	.980	--
G_{st}	36.7	.992	297
G_{wst}	45.1	.990	108

Table 4: Costs and tuning times for Trial 4

The results from these four trials were shown because they illustrate the variety of results that were seen from using model synthesis weighting. Some weighting schemes reduced the initial cost for model tuning by as much as 75%. However, the improved initial cost does not necessarily translate into reduced tuning times or lower cost final identified models. This is a result of the fact that the non-convex model tuning is not a direct path to a minimum solution. Varying the initial model, through various weightings in the synthesis step, can cause the optimization to find a different local minimum for the cost function or it can cause the optimization to take a different path to the same solution.

In most cases, using a proper input-output weighting scheme in both the synthesis and tuning steps resulted in higher average FRAC values, but at the expense of higher final cost function values. This suggests that the final models have improved correlation for the lower response input-output pairs.

V. Conclusion

System identification tools can use weighting schemes to improve accuracy for complex models. Weightings are used to even out the magnitude discrepancies between transfer function poles and zeros, as well as input-output pairings. Simple weighting schemes were shown to reduce cost function values by as much as 75%. It was shown that improvement in the initial cost function does not guarantee faster tuning or more accurate final models. It was also shown that weighting in both the synthesis and tuning steps created models with improved correlation, often at the expense of higher cost function values.

VI. Future Work

Future work includes deriving the analytical gradients for a variety of cost functions. This would allow the user to tune their model to the cost function that best meets their end requirements. In addition to the gradients, the Hessians of the cost functions can be calculated to further reduce the tuning time. Finally, these methods are well suited for implementation in a distributed computing framework.

In this work, models were generally synthesized with low order estimates to keep tuning times at a reasonable working length. Another avenue for future work is to synthesize and tune models with a more appropriate number of states. This will increase tuning time, but it may also alleviate the problems with encountering local minima in the tuning function.

Finally, we used the tri-diagonal approach to parameterize the system. This over-parameterizes the system. There are other parameterizations [8,11] that may be less complex and more numerically robust. These alternate parameterizations may also entail re-computation of the cost function gradient.

References

1. Jacques, R., Liu, K, Miller, D., *Identification of Highly Accurate Low Order State Space Models in the Frequency Domain.* Signal Processing, 1996. **52**: p. 195-207.
2. Van Overschee P., D.M.B., *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. 1996: Kluwer Academic Publishers.
3. Juang, J., *Applied System Identification*. 1994, Englewood Cliffs, NJ: Prentice Hall.
4. McKelvey, T., *Frequency Weighted Subspace Based System Identification in the Frequency Domain*, in *34th IEEE Conference on Decision & Control*. 1995, IEEE: New Orleans, La.
5. Liu, K., Jacques, R., Miller, D., *Frequency Domain Structural System Identification by Observability Range Space Extraction.* Journal of Dynamics Systems, Measurement, and Control, 1996. **118**: p. 211-220.
6. Lacy, S., Babuska, V., *Input-Output Data Scaling for System Identification*, in *14th IFAC Symposium on System Identification*. 2006, IFAC: Newcastle, Australia.
7. Babuska V., C.D., Lane S, Lacy S., *FRF Correlation and Error Metrics for Plant Identification*, in *Structures, Structural Dynamics, and Materials Conference*. 2005: Austin, TX.
8. Chen, C.-T., *Linear Systems Theory and Design 3rd edition*. 1999, New York: Oxford University Press.
9. McKelvey T., F.A., Moheimani R., *Subspace-Based System Identification for an Acoustic Enclosure.* Journal of Vibration and Acoustics, 2002. **124**: p. 414-419.
10. Lacy S., B.V., Schrader K., Fuentes R., *System Identification of Space Structures*, in *ACC*. 2005: Portland, OR.
11. McKelvey T., H.A., Ribarits T., *Data Driven Local Coordinates for Mulitvariable Linear Systems and Their Application to System Identification.* Automatica, 2004. **40**: p. 1629-1635.