SENSITIVITY ANALYSIS OF CLOSED-LOOP CONTROLLER PERFORMANCE: A MODEL-BASED BOOTSTRAP APPROACH

Michael Rotkowitz [1]

Abstract: We present a new framework, using a model-based bootstrap with pre-whitening and post-blackening, to analyze the robustness of overall controller synthesis methods to model uncertainties. This considers both the system identification and the subsequent controller synthesis together as an overall synthesis method to be analyzed. This framework is extremely flexible, can be used with any identification or synthesis method, and can easily be extended to many other types of problems. As a very encouraging first step, we see a dramatic confirmation of the dangers associated with certain synthesis methods.

Keywords: Bootstrap, robust control, system identification, $\mathcal{H}_2$ control, $\mathcal{H}_\infty$ control, identification for control

1. INTRODUCTION

This paper proposes a framework for studying the robustness of controller design. The framework is geared for a great deal of flexibility, so that the system identification and the controller synthesis methods may be jointly analyzed, so that any such methods can be studied, and with an eye towards more complicated synthesis problems where other tools may not be available, such as decentralized identification and control, or nonlinear control.

The stage of the research presented is preliminary, and several aspects of the procedure are fairly heuristic. The preliminary results are very promising however, and this combined with the flexibility offered suggest that this may be a worthwhile avenue for future research.

1.1 Prior Work

As the bootstrap celebrates its twenty-fifth anniversary (Efron, 1979), it has revolutionized data analysis in many fields. However, it has been slow to work its way into the engineering literature, and in particular, into systems and control theory.

A high-level introduction to the bootstrap was written for the signal processing community, in (Zoubir and Boashash, 1998). Most mentions of the bootstrap in the controls literature pertain to system identification. A survey paper of estimation methods in paper machining (Wang et al., 1993) lists several bootstrap applications. The bootstrap was used to estimate the variance of undermodeled system models, or models which are not flexible enough to describe the underlying systems, in (Tjarnstrom and Ljung, 2002), with the basic conclusion that the estimates were in agreement with Monte Carlo simulations. It was used for improved tests of rank determination in (Camba-Mendez and Kapetanios, 2004).

A rare attempt to use the bootstrap in controller synthesis can be found in (Aronsson et al., 1999).

[1] Automatic Control
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
michael.rotkowitz@s3.kth.se

2. PRELIMINARIES

2.1 Controller synthesis

The standard two-input two-output framework is illustrated in Figure 1.
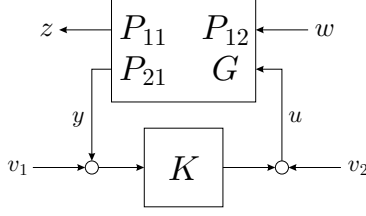

Fig. 1. Linear fractional interconnection of P and K

The objective is to find K such that the mapping from w to z is as small as possible, where small may be defined in many ways, usually as one of several system norms. When these signals as infinite-dimensional, as they often are, we must also ensure that all nine of the maps in the diagram, from (w, v_1, v_2) to (z, u, y), are all stable.

The generalized plant P is appropriately partitioned into four subblocks, and the P_{22} block which takes control inputs to sensor measurements we represent as G. When the plant and controller are assumed to be linear, this optimal control problem may be written as follows.

$$
\begin{aligned}
\text{minimize} \quad & \|P_{11} + P_{12}K(I - GK)^{-1}P_{21}\| \\
\text{subject to} \quad & K \text{ stabilizes } P
\end{aligned}
\tag{1}
$$

2.2 System identification

In this framework, the system identification problem is to estimate G given measurements of u and y. This is an estimation problem, so it is not surprising that the bootstrap has begun to emerge here sooner than other areas of control. However, it is important to note that the objective of estimation may be fundamentally different when the ultimate goal is estimation for the purpose of control (Hjalmarsson *et al.*, 1994; Lindqvist and Hjalmarsson, 2001).

This paper addresses this issue by using bootstrap techniques to analyze the variability of the system identification method together with the controller synthesis method.

2.3 Data sets

This section gives a brief description of the data sets considered in this paper.

2.3.1. Hair Dryer This data is taken from the Matlab system identification toolbox, and is from a process similar to a hair dryer. Air is fanned through a tube and heated at the inlet, with the input being the power of the heating device, and the output is the outlet air temperature. The data is shown in Figure 2 with the input on the bottom, the output on the top, and the means removed.
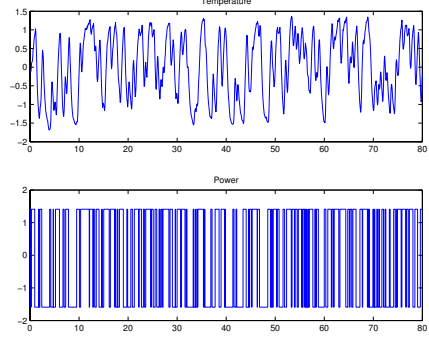

Fig. 2. Hair Dryer Data

2.3.2. DC Motor This data is taken from an older version of the Matlab system identification toolbox, and is from a simulated DC motor. The input is a voltage, and while the output in the toolbox is both an angular position and velocity, we take the output to be just the position, so that we can observe what happens when a controller can only access part of the state of the system it's controlling. The detrended data is shown in Figure 3.
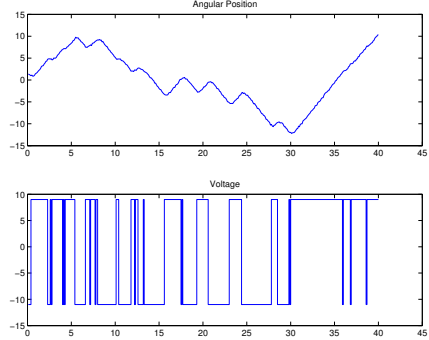

Fig. 3. DC Motor Data

3. BOOTSTRAPPING TECHNIQUES

The standard bootstrap, and much of the discussion in (Efron and Tibshirani, 1993) applies only to independent data. Extensions have been made to dependent data such as time series over the past decade and a half, and some of these techniques are discussed in (Davison and Hinkley, 1997). This section discusses how to further extend these techniques to input-output time series such as the data considered here. Clearly, naively bootstrapping input-output pairs would result in whitening and

obliterate the relationship of the data if there is any delay between the input and the output. It seems likely that a simple extension of the block bootstrap technique to input-output data, that is, randomly choosing blocks of time for which both the input and output are taken, would also result in excessive whitening since the input at the end of each block would be affecting an output from a different block. We consider two of the techniques outlined in (Davison and Hinkley, 1997) to avoid whitening in time series, their possible extensions to I-O data, and analogies with bootstrapping regressions.

3.1 Blocks of blocks

With this method, each block is considered separately when calculating the statistic of interest. This bootstrapping technique itself would extend seamlessly to I-O data, as we could just take the input and output data for each block. It's a very appealing option, since the inputs at the end of each block and the outputs at the beginning of each block would no longer erroneously enter into coefficient estimation. The drawback seems to be that since this does not actually produce a new series, it may not be clear how to extend the calculation of the statistic, usually a function of the time series, to the block of blocks. This wasn't a problem for the AR(1) example worked out in (Davison and Hinkley, 1997), but it's non-trivial for the statistics we would like to study, namely, closed-loop performance.

3.2 Pre-whitening and post-blackening

With this method, a model is fit to the time series, and then the residuals, which have now been pre-whitened, are subject to a block bootstrap. This is an appealing compromise between bootstrapping residuals and blocks in that it attempts to remove trends which will be broken by the resampling, but then acknowledges that some trends may remain. It also extends easily to I-O data, as we can model the output as a function of the input, then block bootstrap the residuals, or innovations, and then regenerate the output time series with the model and the new set of innovations.

3.3 Comparison

Since the input remains fixed in the latter technique, it seems more appealing for cases where the input is chosen by the experimenter, rather than being observed, just as bootstrapping residuals for regression analysis is reasonable when the independent variables are chosen. The blocks of

blocks technique seems more appropriate when there is randomness associated with the input as well as the output, just as bootstrapping pairs is more often used for regression analysis when the independent variable is observed. This would be one argument for using blocks of blocks with the hair dryer data and DC motor data. However we would like to develop a robust computational framework in which different system identification methods, controller synthesis methods, and performance measures can be easily swapped, and so the reconstruction of a new output series for each bootstrap sample with whitening and blackening is desired, and the need to adapt different functions of time series to blocks of blocks is prohibitive. Moreover, in many cases we do get to choose the input data for system identification, and the main contribution of this paper is intended to be the introduction of an extremely flexible technique, not to resolve anything about these particular data sets. We thus use pre-whitening and post-blackening.

The residuals were bootstrapped such that the block length was always equal to the order of the model fit during the pre-whitening. Trying out other, probably longer block lengths, would be a worthwhile future step.

It should be noted that despite the the different justification for each, bootstrapping pairs or residuals for regressions often produces very similar results.

4. MAIN PROCEDURE

Given input-output data for a system, a method for system identification, a procedure for synthesizing an optimal (in some sense) controller, and a measure of performance, we apply the following. When constructing P from G at any stage, we are referring to P as the generalized plant and to G as the lower right-hand block P_{22} as in Figure 1, and we construct the rest of P so as to minimize $z = \begin{bmatrix} x \\ \eta u \end{bmatrix}$. Thus what we are attempting to keep small is a vector of the state and the input, and η trades off their relative importance.

- Estimate $\hat{G}$ from $\begin{bmatrix} y & u \end{bmatrix}$, the original data.
- Construct $\hat{P}$ from $\hat{G}$.
- Synthesize K based on $\hat{P}$.
- For each b in 1:B
  · Generate y^{*b} from $\hat{G}$ with pre-whitening and post-blackening.
  · Estimate G^{*b} from $\begin{bmatrix} y^{*b} & u \end{bmatrix}$, the bootstrapped data.
  · Construct P^{*b} from G^{*b}.
  · Find $\hat{\theta}^{*b}$, the performance of K closed around P^{*b}.

In words, we first perform system identification on the original data to find $\hat{G}$, construct the generalized plant $\hat{P}$, then synthesize optimal controller K based on that model. Then for each bootstrap repetition, we construct a new output time series using pre-whitening and post-blackening as described above, find a new model G^* for the system with the same method on the new data, construct P^*, and measure the performance of the system with K closed around P^*.

This was done for $B = 1000$ bootstrap samples on the dryer and motor data for each of two different identification methods and two different synthesis methods. The performance measure was fixed as the $\mathcal{H}_\infty$-norm, and evaluated with dhfnorm() from the μ-Analysis and synthesis toolbox. Plots were constructed for each of the eight combinations showing the 1000 bootstrapped outputs, along with the original data, but due to their enormous size, they have been omitted.

4.1 System identification methods

The two system identification functions used are pem(), a prediction error method from the Matlab system identification toolbox, and n4sid(), a subspace method from the same toolbox. Both are relatively 'black-box' methods which estimate the order of the model as well as the coefficients. Since these and other methods can of course fit the data points well, a big part of this contribution is to present a framework to analyze the ultimate efficacy of these methods after they are used to not only estimate a model, but to find a controller based on that model and connect it to the plant. In some sense, we will also demonstrate the danger of using black-box methods in general.

4.2 Controller synthesis methods

The first synthesis method used is minimization of the $\mathcal{H}_2$-norm, or LQG control, and is interpreted as minimizing the RMS error of the output subject to a white noise input. This was done with dh2lqg() from the Robust control toolbox. The second method minimizes the $\mathcal{H}_\infty$-norm, which is also the norm which we will use to measure performance on each bootstrap sample for all synthesis methods. This is done using dhinfopt(), also from the Robust control toolbox.

5. RESULTS

For each set of data and estimation method, we present a pair of histograms, one for each synthesis method. On the x-axis is the log base 10 of the closed loop $\mathcal{H}_\infty$-norm, for each of the samples

where the closed-loop was still stable. All were obtained using $\eta = 1$, so that the size of the state and input were penalized equally. Each plot title indicates the data set, the sysid method, the controller synthesis method, and how many of the 1000 were unstable. The dashed line corresponds to the norm with the original data, consistent with the convention in the bootstrap literature.

It's clear that for the first pair, shown in Figure 4, both synthesis methods have yielded similar controllers, as the histograms and number of unstable instances are nearly identical.

In general, there seems to be less variability associated with the performance when the subspace estimation method is used rather than the prediction error method.

We also see that the closed-loop performance for the hair dryer data exhibits a bimodal nature, while that of the DC motor does not.

For the hair dryer data, we had full state feedback, and thus guaranteed stability margins for the controllers derived with $\mathcal{H}_2$ synthesis. We see that about 3% of the bootstrap samples had an unstable closed-loop when the prediction error method was used for identification, and less than 1% when the subspace method was used, and that this was nearly identical whether the controller was synthesized to optimize the $\mathcal{H}_2$-norm or the $\mathcal{H}_\infty$-norm.

For the DC motor data, however, we did not have full state feedback. It was shown in (Doyle, 1978) that one could find the controller which would minimize the closed-loop $\mathcal{H}_2$-norm, and the closed-loop could still be arbitrarily close to unstable. It was soon after suggested that minimizing the $\mathcal{H}_\infty$ norm would be a more sensible thing to do in many cases.

Looking at the third pair of histograms in Figure 6, where the prediction error method is used on the DC motor data, it seems that we have encountered an example of such a synthesis problem. In the histogram on the left side of this figure, we first see a dramatic confirmation of the LQG danger. The controller is found to minimize the $\mathcal{H}_2$-norm, and a whopping 303 out of the 1000 bootstrap samples have an unstable closed loop.

We then see a confirmation of the increased robustness associated with using $\mathcal{H}_\infty$ synthesis in these situations as we look to the histogram on the right side of this figure. Given the same input-output data and the same sysid method, but finding the controller to minimize the $\mathcal{H}_\infty$-norm, only 2 of the bootstrap samples yield an unstable closed-loop, and the histogram is well behaved otherwise.
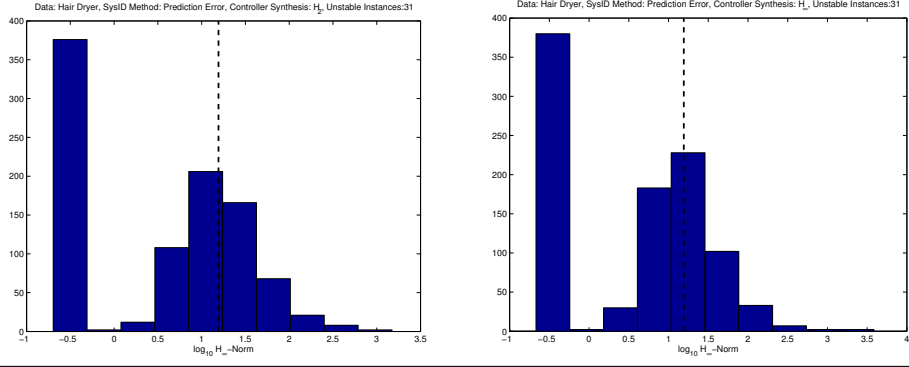
Fig. 4. Hair dryer data with prediction error method, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis
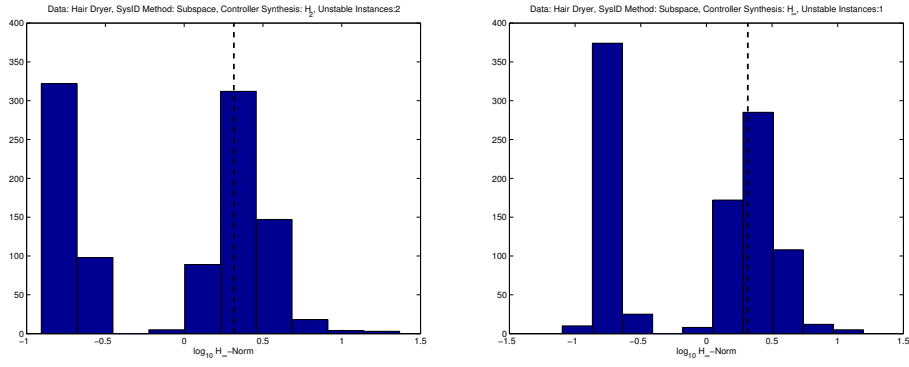

Fig. 5. Hair dryer data with subspace method, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis
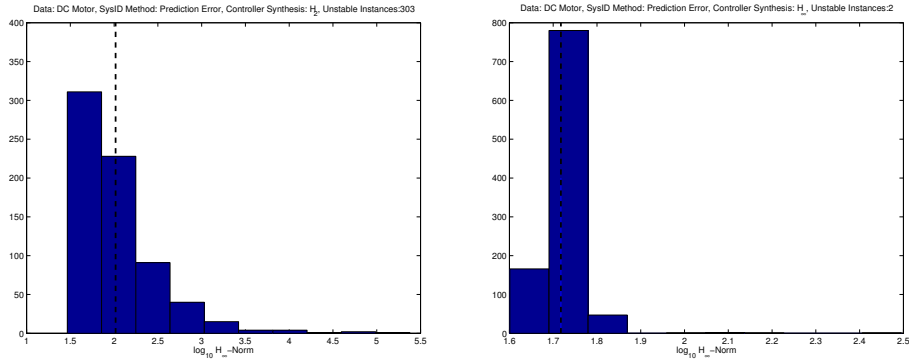

Fig. 6. DC motor data with prediction error method, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis
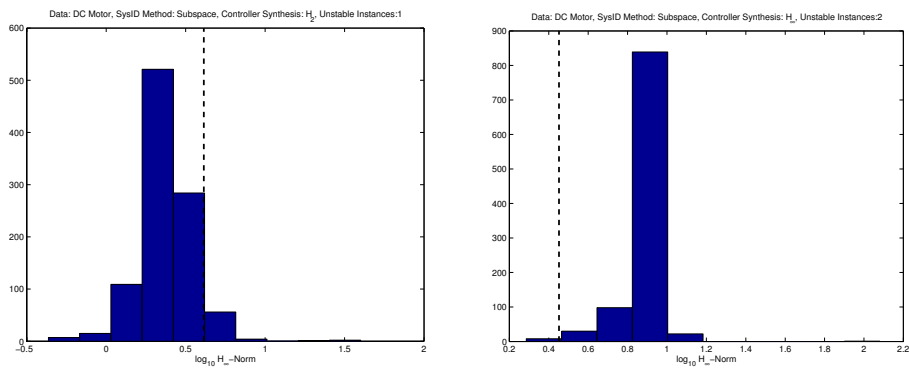

Fig. 7. DC motor data with subspace method, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis

This provides a nice sanity check, showing that our traditional notions of robustness are consistent with those explored here with our histograms. It also elucidates the possibility that when this framework is applied to more complex problems, other robustness pitfalls which we do not yet know of may be identified.

6. FUTURE WORK

6.1 Adjustments within framework

Several tweaks to the procedure used here would be worth exploring. In particular, the same estimation technique was always used to fit the initial model that the controller was synthesized around, and to estimate the model of each of the bootstrap samples. It may be more realistic to carefully pick the initial model, then fit larger models on the bootstrap samples.

6.2 Bagging

The surprising variability of closed-loop performance after the controller is optimized for one set of data suggests that a technique like bagging may be beneficial to synthesize more robust controllers. This would involve synthesizing a controller based on each bootstrap sample, in other words, performing system identification and controller synthesis for each artificial set of input-output data, and then averaging the controllers in the appropriate sense.

6.3 Theory

The bootstrap is particularly useful for generating standard deviations and confidence intervals for statistics where no other method is known. While we are able to study the variability of the closed-loop performance, the theory which allows confidence intervals to be easily deduced from such histograms may break down when the statistic of interest can take on infinite values, which is what happens when the closed loop goes unstable. Adapting this theory so that confidence intervals could be obtained for closed-loop performance may be a worthwhile step.

7. CONCLUSIONS

We have presented a framework to analyze the robustness of overall controller synthesis methods to model uncertainties. We discussed several methods of bootstrapping dependent data, and developed a procedure based on pre-whitening and post-blackening to generate bootstrap samples of system identification data. The variability of closed-loop performance was then observed, and we saw a confirmation of the dangers associated with certain synthesis methods. Most important, the flexibility provided by this framework suggests that it could easily be extended to far more complicated problems.

8. ACKNOWLEDGMENTS

REFERENCES

Aronsson, M., L. Arvaston, J. Holst, B. Lindoff and A. Svensson (1999). Bootstrap control. In: *Proceedings of the American Control Conference.* pp. 4516–4521.

Camba-Mendez, G. and G. Kapetanios (2004). Bootstrap statistical tests of rank determination for system identification. *IEEE Transactions on Automatic Control* **49**(2), 238–243.

Davison, A.C. and D.V. Hinkley (1997). *Bootstrap methods and their application.* Cambridge University Press.

Doyle, J. (1978). Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control* **23**(4), 756–757.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. Vol. 7. pp. 1–26.

Efron, B. and R. Tibshirani (1993). *An Introduction to the Bootstrap.* Chapman and Hall/CRC.

Hjalmarsson, H., M. Gevers, F DeBruyne and J. Leblond (1994). Identification for control: closing the loop gives more accurate controllers. In: *Proceedings of the IEEE Conference on Decision and Control.* pp. 4150–4155.

Lindqvist, K. and H. Hjalmarsson (2001). Identification for control: Adaptive input design using convex optimization. In: *Proceedings of the IEEE Conference on Decision and Control.* pp. 4326–4331.

Tjarnstrom, F. and L. Ljung (2002). Using the bootstrap to estimate the variance in the case of undermodeling. *IEEE Transactions on Automatic Control* **47**(2), 395–398.

Wang, X.G., G.A. Dumont and M.S. Davies (1993). Estimation in paper machine control. *IEEE Control Systems Magazine* **13**(4), 34–43.

Zoubir, A.M. and B. Boashash (1998). The bootstrap and its application in signal processing. *IEEE Signal Processing Magazine* **15**(1), 56–76.